KERNFORSCHUNGSANLAGE JÜLICH GmbH

Institut für Festkörperforschung

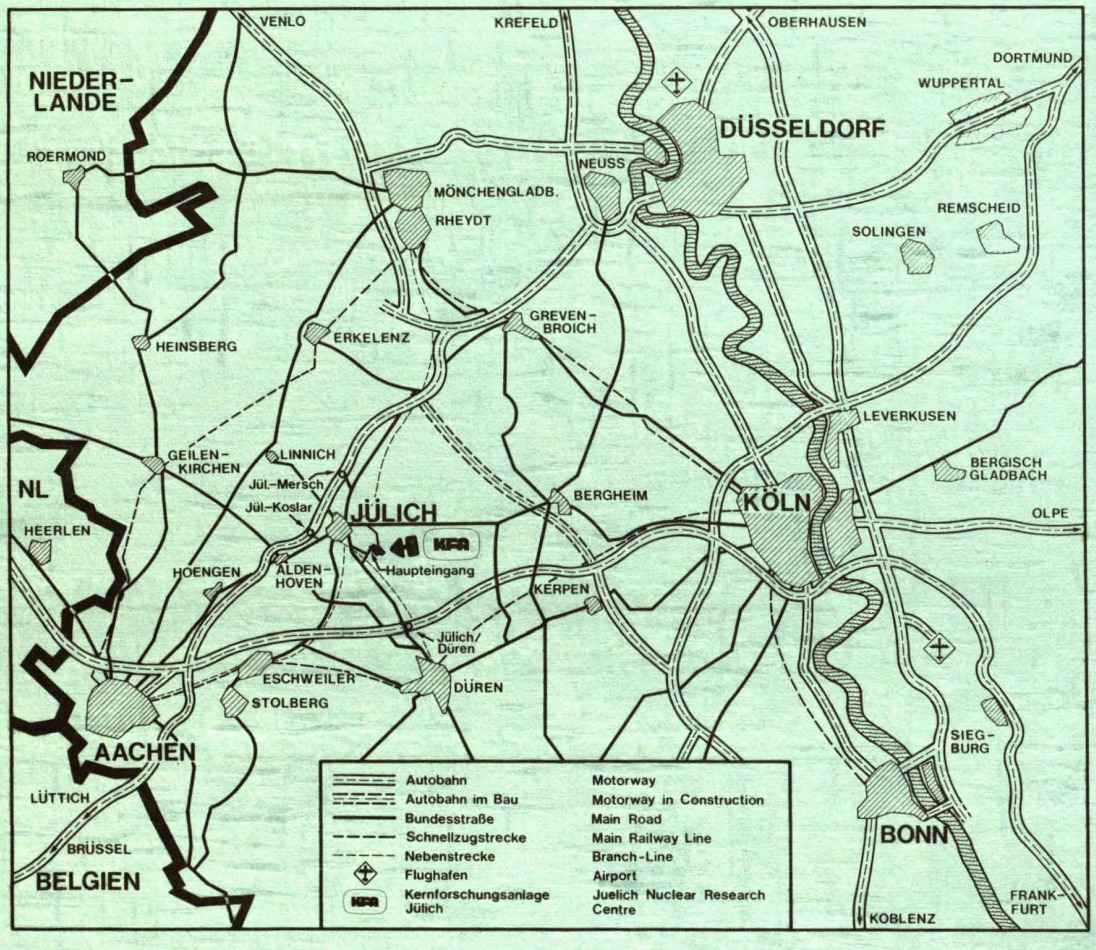
A Detailed Survey of Numerical Methods for Unconstrained Minimization

Part 1: Conjugate Direction and Gradient Methods

by

K. Mika, Th. Chaves

Jül-1643
Januar 1980
ISSN 0366-0885



Als Manuskript gedruckt

Berichte der Kernforschungsanlage Jülich - Nr. 1643

Institut für Festkörperforschung Jül - 1643

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH,
Jülich, Bundesrepublik Deutschland

A Detailed Survey of Numerical Methods for Unconstrained Minimization

Part 1: Conjugate Direction and Gradient Methods

by

K. Mika and Th. Chaves*

*Pontifícia Universidade Católica, Rio de Janeiro

Summary

A detailed description of numerical methods for unconstrained minimization is presented. This first part surveys in particular conjugate direction and gradient methods, whereas variable metric methods will be the subject of the second part. Among the results of special interest we quote the following. The conjugate direction methods of Powell, Zangwill and Sutti can be best interpreted if the Smith approach is adopted. The conditions for quadratic termination of Powell's first procedure are analyzed. Numerical results based on nonlinear least squares problems are presented for the following conjugate direction codes: VA04AD from Harwell Subroutine Library and ZXPOW from IMSL, both implementations of Powell's second procedure, DFMND from IBM-SLMATH (Zangwill's method) and Brent's algorithm PRAXIS. VA04AD turns out to be superior in all cases, PRAXIS improves for high-dimensional problems. All codes clearly exhibit super-linear convergence. Akaike's result for the method of steepest descent is derived directly from a set of nonlinear recurrence relations. Numerical results obtained with the highly ill conditioned Hilbert function confirm the theoretical predictions. Several properties of the conjugate gradient method are presented and a new derivation of the equivalence of steepest descent partan and the CG method is given. A comparison of numerical results from the CG codes VA08AD (Fletcher-Reeves), DFMCG (the SSP version of the Fletcher-Reeves algorithm) and VA14AD (Powell's implementation of the Polak-Ribière formula) reveals that VA14AD is clearly superior in all cases, but that the convergence rate of these codes is only weakly superlinear such that high accuracy solutions require extremely large numbers of function calls.

Contents

1. Introduction 1

2. One-dimensional minimization 5

2.1 Grid search 6

2.2 Sequential search techniques 6

2.2.1 Equally spaced search 7

2.2.2 Dichotomous search 8

2.2.3 Fibonacci search 8

2.2.4 Golden-section search 15

2.3 Function approximation techniques 17

2.3.1 Quadratic interpolation 18

2.3.2 Cubic interpolation 22

2.3.3 A line search strategy using Davidon's technique 25

3. Direct search methods 28

3.1 The simplex method 29

3.2 Rosenbrock's method 33

4. Conjugate direction methods 37

4.1 Conjugate directions 37

4.2 The method of Smith 45

4.3 The method of Powell 47

4.3.1 Powell's first procedure 47

4.3.2 Powell's second procedure 53

4.4 The method of Zangwill 58

4.5 On quadratic termination of Powell's modified first procedure 63

4.6 Some other algorithms without derivatives ... 66
4.6.1 The method of Brent 66
4.6.2 The method of Sutti 69
4.6.3 The method of Brodlie 71
4.7 Some numerical results 73

5. Gradient methods 76

5.1 The method of steepest descent 76
5.1.1 An example 78
5.1.2 The single-step convergence rate 80
5.1.3 Upper bound for the single-step convergence rate 82
5.1.4 The asymptotic behaviour of the optimum gradient method 86
5.1.5 Some numerical results 94
5.2 The method of conjugate gradients 95
5.2.1 A description of the method 96
5.2.2 An important orthogonality relation .. 98
5.2.3 Determination of the coefficients β_{ki} 99
5.2.4 Other approaches to the method of conjugate gradients 101
5.2.5 The reduction of $\| x_k - x^* \|$ 104
5.2.6 The method of conjugate gradients as an optimal process 106
5.2.7 Quadratic termination without perfect line searches 109
5.2.8 The method of Shah, Buehler and Kempthorne 112
5.2.9 Some numerical results 115

References 120

Acknowledgment 126

1. Introduction

Numerical methods for unconstrained minimization have a large field of applications, which can be divided into two classes: problems which can only be solved by these methods, and problems which can be solved by especially designed methods but where methods for unconstrained minimization are also applicable. Furthermore, methods for unconstrained minimization often form a substantial part of algorithms for constrained optimization. Examples for the first class are: Minimization of integrals from variational problems which can be reduced to functionals with a finite number of parameters, maximization of determinants in order to find optimal measurement points according to the statistical theory of experimental design, or estimation of parameters if the maximum likelihood criterion is applied. The second class mainly consists of problems which can be formulated as the minimization of a sum of squares of residuals, like all nonlinear least squares problems or, for the particular case that all residuals can vanish, the problem of solving systems of nonlinear equations. On the other hand, a general minimization problem might also be solved by applying an algorithm for the solution of nonlinear equations to the set of components of the gradient, as a result any stationary point will be found but not necessarily the actual minimum. It must be stressed, however, that the methods to be described here will only find local minima, whereas the problem of finding the global minimum even in a restricted domain can in general not be solved.

Optimization problems are usually highly complex, and computer codes - especially for constrained problems - reflect this complexity to a large extent. These codes must not only be based on a safe theoretical ground, but also require a great deal of experimentation with simulated and real life problems of varying degree of difficulty. It is therefore not surprising that excellent algorithms have been designed and developed in research centers (Harwell and Argonne) or in specially founded

institutions like the Numerical Optimization Center of Hatfield. The importance of these methods for large scale experimental research is underlined by the CERN development of the minimization package MINUIT, and a package of computer codes, MINPACK, is under preparation at Argonne.

In the following chapters we give a rather detailed survey of numerical methods for unconstrained minimization. We shall exclude convergence properties of minimization algorithms, also the techniques of matrix factorization originally introduced by Gill and Murray will not be discussed. Our notation tries to adhere to the standard conventions: lower case roman letters for column vectors, lower case greek letters for scalars and upper case roman letters for matrices. If not otherwise stated, we denote by d a general direction, and by p a conjugate direction. Quadratic functions are always quadratic forms with a positive definite matrix G. $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $h : \mathbb{R} \to \mathbb{R}$ describes the dependence of the objective function on the step-width λ along a search direction, α_k denotes the step-width which corresponds to the minimum of h for iteration $k+1$, the minimum of f is denoted by x^*.

Chapter 2 summarizes techniques for one-dimensional minimization, such as golden-section and Fibonacci search applicable if the function is known to be convex, and standard methods of interpolation as used in algorithms for unconstrained minimization. Davidon's cubic interpolation formula is derived explicitly and different representations are evaluated. A complete line search strategy is given at the end of this chapter.

In chapter 3 the direct search or stepping methods are very briefly discussed, and details are given only for the simplex method and Rosenbrock's method. For the latter method a criterion is established which ensures linear independence of the set of search directions.

Conjugate directions, and algorithms based on conjugate directions only, i.e. which do not use any derivatives, are the subject of chapter 4. A unifying approach, which originates from the first algorithm of this kind by Smith, is adopted. Examples illustrate the performance of Powell's first and second procedure and Zangwill's method when applied to quadratic functions. Details are presented which elucidate the quadratic termination property of Powell's first procedure. Four computer codes are compared when applied to nonlinear least squares problems of moderate to high degree of difficulty. The Harwell subroutine VA04AD comes out very favourably. All codes clearly exhibit superlinear convergence.

Gradient methods are defined as those algorithms which use the gradient primarily to build up a search direction - they also use the gradient for the line search - but which do not attempt to approximate the matrix of second derivatives as will be the case with variable metric methods. In chapter 5, the method of steepest descent is treated in detail. For quadratic functions we show that the single step convergence rate of this method is monotonously increasing for each iteration, only in two dimensions this rate is constant. A completely new derivation of Akaike's result is given which states that only two directions determine the search asymptotically for large k . The behaviour of the method of steepest descent is illustrated for a highly ill conditioned example and its intolerably slow convergence rate becomes apparent.

The method of conjugate gradients, another class of gradient methods, is also presented in chapter 5. Among the properties outlined we only mention the interpretation of this method as an optimal process. A new proof of the equivalence of steepest descent partan and the method of conjugate gradients is given. Numerical results for three available CG codes reveal that the Harwell routine VA14AD is superior, however none of these codes clearly shows superlinear convergence.

The second part of this survey is completely concerned with variable metric methods. Chapter 6 deals with those methods which had greatest influence in theory and practice for the whole field of unconstrained minimization: The Davidon-Fletcher-Powell (DFP) method, the symmetric rank one (SR1) formula and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. The property of quadratic termination is shown explicitly for the DFP formula in a way that this proof can be generalized to a large class of updating formulae defined in paragraph 6.2. By stability we understand - as is now common use - the property that the approximation H_{k+1} to the inverse Hessian matrix is positive definite if H_k is positive definite. The conditions for stability of the three methods of this chapter are investigated.

Chapter 7 is devoted to families of variable metric methods: the symmetric one parameter Broyden family, the symmetric two parameter Huang family, the Oren-Luenberger class and Greenstadt's variational family. Of special interest is the Huang family with $\rho = o$: this class of algorithms generates conjugate directions without perfect line searches and can be considered as a *batch processing algorithm* as against the usual VM methods which in this context can be classified as *sequential processing algorithms*, an analogy which stems from Kalman filtering theory.

In the final chapter many aspects of Davidon's optimally conditioned algorithm will be discussed. A direct proof, which avoids projection matrices, is given for the property of quadratic termination. The optimally conditioned parameter turns out to be a special solution of a quite general functional equation for Fletcher's parameter Φ , if self-duality is imposed. The product form of the updating formula is derived explicitly by solving a system of three nonlinear equations originally set up by Spedicato. Numerical results obtained with the computer codes DRVOCR (with derivatives) and OCOPTR (without derivatives) are compared with the corresponding codes VA09AD, VA13AD on the one hand, VA10AD and ZXMIN on the other. The influence of the initial choice H_o is investigated in detail and it turns out that a diagonal matrix accounting for approximate scaling already improves the performance considerably.

2. One-dimensional minimization

To find the minimum of a function with one variable turns out to be an important subproblem in higher-dimensional minimization problems. In this context it is usually formulated as the following problem:

$$\text{Minimize} \quad f(x_k + \lambda p_k) \quad \text{with respect to} \quad \lambda \ , \qquad (2.1)$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is the objective function to be minimized, $x_k \in \mathbb{R}^n$ is the approximation of the minimum x^* after k iterations, p_k is the $(k+1)$-st search direction, and λ is a scalar parameter defining the line $x_k + \lambda p_k$. Relation (2.1) requires to find the *minimum along the line*, also known as the *line search problem*. We say that the line search is perfect, or exact, if this minimum is found. The corresponding value of λ will be denoted by α_k . If more than one minimum along the line can occur, additional requirements will specify the desired minimum, e.g. the first encountered for $\lambda > 0$, or the global one.

Minimizing functions with one variable has also become an important subject in its own right and many methods have been developed. However, a clear distinction between the methods solving problem (2.1), and those solving the general one-dimensional minimization problem must be made, as we shall see.

Whereas in (2.1) a rough estimate of a *local minimum* - and very often even any λ satisfying $f(x_k+\lambda p_k) < f(x_k)$ - will be accepted as an approximate solution to the posed problem, a solution to the general one-dimensional minimization problem will only be acceptable, if the minimum is located within a prescribed accuracy. If more than one minimum exists, normally the *global minimum* is of interest, although situations may occur where all minima in a given interval are required.

The methods to be considered here can be divided into two groups: those based on a *comparison of function values*, appropriate for particular classes of functions, and those *approximating the function* by some low order polynomial. A combination of these two approaches is discussed briefly at the end of this chapter.

2.1. Grid search

The easiest way to locate a minimum - and sometimes the most efficient if e.g. several minima in a given interval can occur - is the grid search: Divide the interval under consideration with length L into k equally spaced intervals and select the point with the smallest function value on this grid as the optimum. Compared with other methods, this technique is extremely simple and absolutely robust. The accuracy of the minimum position is $2L/k$ after $k+1$ function evaluations. This method can only be used if the required accuracy is not too high, say 10^{-5}, and prohibits itself for higher-dimensional problems. However, a good overall impression of the function behaviour will be obtained. If on the other hand the rather restrictive assumption of unimodality is made, some highly efficient methods are applicable as will be shown next.

2.2. Sequential search techniques

As before, these methods only require function values which are to be compared. The class of admissible functions therefore includes even discontinuous functions, solely $f:[a,b] \to \mathbb{R}$ must be uniquely defined in the interval $x \in [a,b]$.

Definition: A function $f:[a,b] \to \mathbb{R}$ is *unimodal* on $x \in [a,b]$, if there exists a unique point $x^* \in [a,b]$ such that $f(x)$ is strictly monotonously increasing with increasing distance $|x-x^*|$ from x^* .

An example given by Brent [8]:

$$f(x) = \begin{cases} 1 - x , & x \leq 0 \\ x , & x > 0 \end{cases}$$

reveals that this unimodal function does not even assume its minimum value, yet all methods to be described in this paragraph will find x^* .

From now on we assume $f(x)$ to be unimodal. Then the following property furnishes the basis of sequential search methods:

If $f(x)$ is unimodal on $[a,b]$, the function values of $f(x)$ at the four points a, b, x_1, x_2, where $x_1<x_2$ are two interior points, are sufficient to bracket the minimum position x^* by the reduced interval I_1:

$$I_1 = [a,x_2], \quad \text{if} \quad f(x_2) \geq f(x_1) , \qquad (2.2a)$$

$$I_1 = [x_1,b], \quad \text{if} \quad f(x_2) < f(x_1) . \qquad (2.2b)$$

To show this property we assume that $f(x_2) \geq f(x_1)$, and $x^*>x_2$. Then $\underline{f}(x^*)<f(x_1)\leq f(x_2)$, where $\underline{f}(x^*)$ denotes the infimum of $f(x)$ (see the example given above). Thus the assumption $x_1<x_2<x^*$ leads to a contradiction of unimodality proving (2.2a). Similarly (2.2b) can be shown.

2.2.1. Equally spaced search

The efficiency of a sequential method is given by its reduction coefficient r_N , where r_N is the length of the interval containing the minimum after N function evaluations (in addition to the two function evaluations at a and b, which in the following will not be counted explicitly) and where the original interval has unit length.

For simplicity let $a=o$ and $b=1$. Then a most natural choice of the interior points x_1, x_2 would be: $x_1=\frac{1}{3}$, $x_2=\frac{2}{3}$.
After the first two calls we have: $r_2=\frac{2}{3}$. Then two new function values must be calculated before a further reduction is possible. This leads to the following sequence:

$$\begin{array}{lccccc} k: & 2 & 4 & \ldots & 2m = N & \\ r_k: & \frac{2}{3} & (\frac{2}{3})^2 & \ldots & (\frac{2}{3})^m = (\frac{2}{3})^{N/2} & . \end{array}$$

2.2.2. Dichotomous search

A more efficient technique is known as dichotomous search (see e.g. Box et al. | 6 |). If ε is a small quantity which may be of the order of the machine precision, we may set

$$x_1 = \frac{1}{2} - \frac{\varepsilon}{2} , \quad x_2 = \frac{1}{2} + \frac{\varepsilon}{2} ,$$

i.e. try to halve each interval as closely as possible. The reduction after k function calls will be

k	r_k
2	$\frac{1}{2} + \frac{\varepsilon}{2}$
4	$\frac{1}{2}(\frac{1}{2} + \frac{\varepsilon}{2}) + \frac{\varepsilon}{2} = \frac{1}{4} + \frac{\varepsilon}{2}(1 + \frac{1}{2})$
.	.
.	.
.	.
2m	$\frac{1}{2^m} + \frac{\varepsilon}{2}(1 + \frac{1}{2} + \ldots + (\frac{1}{2})^{m-1}) = \frac{1}{2^m} + \varepsilon(1 - \frac{1}{2^m})$.

If we ignore the term proportional to ε we find

$$r_N = \frac{1}{(\sqrt{2})^N} , \tag{2.3}$$

as against $r_N = \frac{1}{(\sqrt{1.5})^N}$ from section 2.2.1.

2.2.3. Fibonacci search

The natural question arises whether there is a search yielding the maximum reduction by an optimal choice of x_1, x_2. This is true to a certain extent: If the number N of function evaluations to be taken is known in advance, the *optimal strategy* is the so-called Fibonacci search first derived by Kiefer |27|.

The main drawback of the two sequential methods described above obviously results from the fact, that for each reduction *two*

new function values are necessary. The possibility to reduce the interval already after each new function value finally leads to the optimal search.

Consider iteration i where

$$x_o^{(i)} < x_1^{(i)} < x_2^{(i)} < x_3^{(i)} , \qquad i = 1,\ldots,N-1 .$$

Especially we have

$$x_o^{(1)} = o , \quad x_3^{(1)} = 1 .$$

We set

$$x_3^{(i)} - x_o^{(i)} \equiv \Delta^{(i)}$$

$$x_1^{(i)} - x_o^{(i)} \equiv \alpha_i \Delta^{(i)}$$

$$x_2^{(i)} - x_o^{(i)} \equiv \beta_i \Delta^{(i)} .$$

The optimal choice then depends on three assumptions given below.

Assumption a): $x_1^{(i)}$, $x_2^{(i)}$ are symmetric with respect to the center of $[x_o^{(i)}, x_3^{(i)}]$.

This means that

$$\alpha_i + \beta_i = 1 , \qquad i = 1,\ldots,N-1. \tag{2.4}$$

Consider the next iteration (i+1) . Without loss of generality (because of the symmetry) we assume that $f(x_2^{(i)}) \geq f(x_1^{(i)})$. Then from (2.2)

$$x_o^{(i+1)} = x_o^{(i)} ,$$

$$x_3^{(i+1)} = x_2^{(i)} .$$

Assumption b): Require $x_2^{(i+1)} = x_1^{(i)}$ in order to save one function evaluation.

This leads to the equation

$$x_3^{(i+1)} - x_o^{(i+1)} \equiv \Delta^{(i+1)} = x_2^{(i)} - x_o^{(i)} = \beta_i \Delta^{(i)}$$

$$x_2^{(i+1)} - x_o^{(i+1)} \equiv \beta_{i+1} \Delta^{(i+1)} = x_1^{(i)} - x_o^{(i)} = \alpha_i \Delta^{(i)}$$

Substitution yields

$$\beta_{i+1} \beta_i = \alpha_i \quad , \qquad i = 1,\ldots,N-2 \; . \tag{2.5a}$$

With (2.4) we finally obtain a nonlinear recurrence relation for β_i :

$$\beta_i (\beta_{i+1} + 1) = 1 \; , \qquad i = 1,\ldots,N-2 \; . \tag{2.5b}$$

If β_1 were known, all β_i , $i>1$, would follow from (2.5b). Instead, (2.5b) is solved backwards prescribing β_{N-1} , which yields

Assumption c): $\beta_{N-1} = \frac{1}{2}$, i.e. bisection of the final interval.

This demonstrates why the optimal search requires a fixed number of function evaluations in advance. Strictly speaking, $\beta_{N-1} = \frac{1}{2}$ contradicts the condition $x_1^{(N-1)} < x_2^{(N-1)}$, therefore the final values should be separated by some small ε in the same way, as it was necessary for the dichotomous search.

Let r_j denote the reduction coefficient after j function evaluations.[1] Then we have after two calls $r_2 = \beta_1$, after three calls $r_3 = \beta_2 \beta_1, \ldots$, and after N calls

$$r_N = \beta_{N-1} \beta_{N-2} \cdots \beta_1 \; . \tag{2.6}$$

Especially after the first call there is no reduction: $r_1 = 1$.

---

1) To be more precise, throughout this section r_j also depends on N , the total number of function evaluations to be taken.

The following equations hold:

$$r_i = \beta_{i-1}\, r_{i-1} , \qquad i = 2,\ldots,N \qquad (2.7a)$$

$$r_i = \beta_{i-1}\, \beta_{i-2}\, r_{i-2} , \qquad i = 3,\ldots,N . \qquad (2.7b)$$

Inserting (2.5b) into (2.7b) and using (2.7a) we obtain

$$r_i = (1 - \beta_{i-2})\, r_{i-2} = r_{i-2} - r_{i-1} , \; i = 3,\ldots,N .$$

Instead of solving (2.5b) it is more convenient to solve the linear two-point boundary-value problem for r_i

$$r_{i-2} = r_i + r_{i-1} , \qquad i = 3,\ldots,N , \qquad (2.8a)$$

with

$$r_1 = 1 , \quad r_N = \beta_{N-1}\, r_{N-1} , \text{ where } \beta_{N-1} = \frac{1}{2} . \qquad (2.8b)$$

The final members of the sequence of r_i are therefore:

$$r_{N-1} = 2\, r_N$$
$$r_{N-2} = 3\, r_N$$
$$r_{N-3} = 5\, r_N$$
$$\vdots$$

The FIBONACCI numbers satisfy the relation[1)]

$$F_j = F_{j-1} + F_{j-2} , \quad F_o = 1 , \; F_1 = 1 , \; (F_2=2,\; F_3=3,\ldots)$$

With

$$r_i = F_{N-i+1} \cdot r_N , \qquad i = 1,\ldots,N$$

we can therefore satisfy (2.8a) and the second boundary condition $r_{N-1} = 2r_N$.

---

1) This sequence dates back to 1202, when Leonardo da Pisa, called Fibonacci, described the process of rabbit multiplication (see Wilde [54], pp. 30).

With $r_1 = 1$ we have $r_N = \frac{1}{F_N}$, and therefore

$$r_i = \frac{F_{N-i+1}}{F_N} , \qquad i = 1,\ldots, N . \qquad (2.9)$$

The final call separates the last two points by ε , the true reduction is

$$r_N' = \frac{1}{F_N} + \varepsilon . \qquad (2.10)$$

For a prescribed accuracy δ , N must at least be chosen to satisfy

$$\frac{1}{F_N} + \varepsilon \leq \delta ,$$

or

$$F_N \geq \frac{1}{\delta - \varepsilon} .$$

Since the FIBONACCI numbers allow the representation

$$F_N = \frac{1}{\sqrt{5}} \left\{ \left[\frac{1}{2}(1+\sqrt{5})\right]^{N+1} - \left[\frac{1}{2}(1-\sqrt{5})\right]^{N+1} \right\} , \qquad (2.11)$$

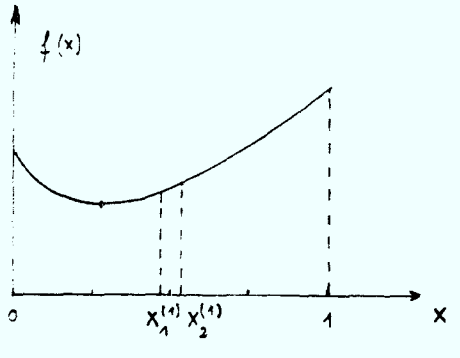
we obtain for large N , neglecting ε :

$$r_N \approx \frac{2\sqrt{5}}{(1+\sqrt{5})} \cdot \frac{1}{\left(\frac{1+\sqrt{5}}{2}\right)^N} \cong \frac{1.3817}{(1.61803)^N} , \qquad (2.12)$$

which in fact is more efficient than the dichotomous search (2.3).

As illustrating examples we discuss the Fibonacci search for N=2, N=3 and N=4 explicitly.

i) <u>N=2</u>

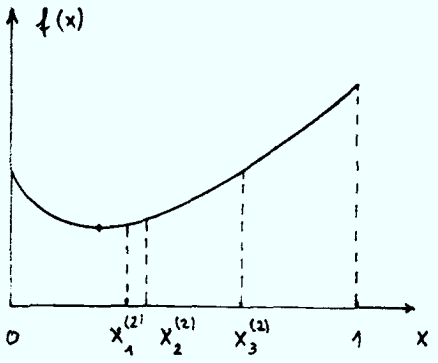


$$x_1^{(1)} = \frac{1}{2} - \frac{\varepsilon}{2}$$

$$x_2^{(1)} = \frac{1}{2} + \frac{\varepsilon}{2}$$

$$r_2' = \frac{1}{2} + \frac{\varepsilon}{2} .$$

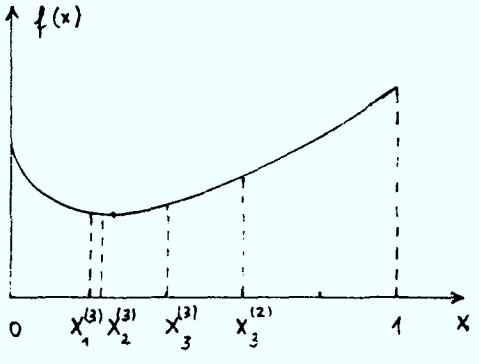
ii) N=3



i:	1	2
$x_0^{(i)}$	0 →	0
$x_1^{(i)}$	$\frac{1}{3}$ →	$\frac{1}{3}$
$x_2^{(i)}$	$\frac{2}{3}$ ↘	$\frac{1}{3} + \varepsilon$
$x_3^{(i)}$	1	$\frac{2}{3}$

$$r_3' = \frac{1}{3} + \varepsilon$$

iii) N=4



i:	1	2	3
$x_0^{(i)}$	0 →	0 →	0
$x_1^{(i)}$	$\frac{2}{5}$ ↘	$\frac{1}{5}$ →	$\frac{1}{5}$
$x_2^{(i)}$	$\frac{3}{5}$ ↘	$\frac{2}{5}$ ↘	$\frac{1}{5} + \varepsilon$
$x_3^{(i)}$	1	$\frac{3}{5}$	$\frac{2}{5}$

$$r_4' = \frac{1}{5}$$

These examples exhibit a certain ambiguity, as to whether we should add ε to r_N or not, depending on the function $f(x)$ at the final outcome. It is, however, easy to account for the finite separation ε *from the beginning*. Assumptions a) and b) still hold, instead of c) we have

Assumption c'): $$r_N = \frac{1}{2} r_{N-1} + \frac{\varepsilon}{2} \,. \qquad (2.13)$$

Equations (2.8) are still valid except for a different β_{N-1}. To find the general solution we again start the sequence

$$r_{N-1} = 2\, r_N - \varepsilon$$

$$r_{N-2} = 3\, r_N - \varepsilon$$

$$r_{N-3} = 5\, r_N - 2\varepsilon$$

$$\vdots$$

leading to

$$r_i = F_{N-i+1}\, r_N - F_{N-i-1}\, \varepsilon \,, \quad i = 1,\ldots, N-1 \,. \qquad (2.14)$$

With $r_1 = 1$ we obtain the final reduction coefficient

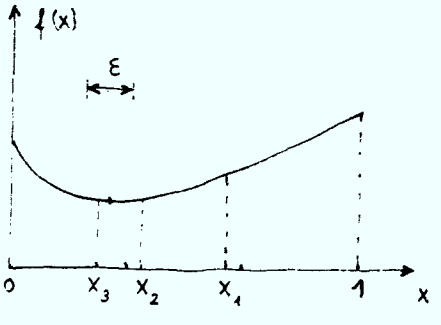
$$r_N = \frac{1}{F_N}\,(1 + F_{N-2}\,\varepsilon). \qquad (2.15)$$

This is the result given by Wilde |54|, p.25, "taking account of the minimum separation ε throughout the search, thus obtaining a final interval of uncertainty slightly smaller than Kiefer's, who does not bother with ε until it is time to place the very last experiment."

Note, however, that (2.1o) is too pessimistic, disregarding the final outcome of $f(x)$, which with probability one half also may give $r'_N = 1 / F_N$, the lowest possible bound.

For $N=3$ and $N=4$ we illustrate Wilde's result. These examples

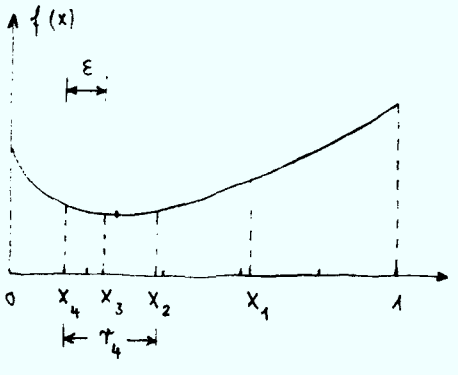
iv) N=3



$$x_1 = \frac{2}{3} - \frac{1}{3}\,\varepsilon$$
$$x_2 = \frac{1}{3} + \frac{1}{3}\,\varepsilon$$
$$x_3 = \frac{1}{3} - \frac{2}{3}\,\varepsilon$$

$$r_3 = \frac{1}{3} + \frac{1}{3}\,\varepsilon$$

v) N=4



$$x_1 = \frac{3}{5} + \frac{1}{5}\,\varepsilon$$
$$x_2 = \frac{2}{5} - \frac{1}{5}\,\varepsilon$$
$$x_3 = \frac{1}{5} + \frac{2}{5}\,\varepsilon$$
$$x_4 = \frac{1}{5} - \frac{3}{5}\,\varepsilon$$

$$r_4 = \frac{1}{5} + \frac{2}{5}\,\varepsilon$$

exhibit in particular, that the first reduction after two calls, x_1 in the examples, is in agreement with Wilde's general formula (equ. (2.13) of [54]):

$$r_2 = \frac{F_{N-1}}{F_N} + \frac{(-1)^N}{F_N}\,\varepsilon \ ,$$

which can be deduced from (2.14/15).

We have not shown that the Fibonacci search is *optimal for given* N , instead we refer to the most elegant but lengthy proof in [54], pp. 25. Two reasons may justify the rather extensive discussion of this method:
i) The Fibonacci search is most interesting in itself;
ii) although not of great practical value because of the limitation to prescribe N in advance, it is fairly easy to obtain a *nearly optimal search without restrictions* from the Fibonacci search as will be shown next.

2.2.4. Golden-section search

Obviously assumption c), the bisection of the final interval, leads to the restriction of the Fibonacci search, i.e. a fixed number of function evaluations. The golden-section search on the other hand does not prescribe the final interval in advance, preserves assumptions a) and b) and requires in addition

Assumption d): β_i is independent of i for all i .

This assumption holds asymptotically for the Fibonacci search, where

$$\beta_i = r_{i+1} \,/\, r_i = F_{N-i} \,/\, F_{N-i+1}$$

is approximately constant for large N and i not too close to N . Assumption d), when applied to (2.5b), yields the equation

$$\beta^2 + \beta - 1 = 0 \ , \qquad (2.16)$$

or

$$\beta = \frac{1}{2}\,(\sqrt{5} - 1) = \frac{2}{(\sqrt{5} + 1)} = 0.618033988... \qquad (2.17)$$

The symmetry relation (2.4) still holds: $\alpha + \beta = 1$. Then (2.16) is equivalent to $1 : \beta = \beta : \alpha$. This is the famous golden section:

The intersections α and β are such that the ratio of length 1 to length β is the same as the ratio of the larger length β to the smaller length α .

The reduction coefficient after N calls follows from (2.6) as

$$r_N = \beta^{N-1} ,$$

i.e. r_N decreases according to a geometrical progression.

We want to compare the efficiency of the golden-section search with that of the Fibonacci search. For this purpose two inequalities are of interest. Let $G_N = \beta^{1-N}$. Then from (2.11) and β from (2.17):

$$F_N = c_N G_N \quad \text{with} \quad c_N = \frac{1}{\sqrt{5}\ \beta^2} \{1 + (-1)^N\ \beta^{2N+2}\} ,$$

where $c_1 = 1$, $c_2 = 1.236...$, and

$$1 < c_N < c_2 , \qquad N \geq 3 .$$

This leads to the inequality $F_N > G_N$ for $N \geq 2$. Further

$$G_N = \beta\ G_{N+1} \text{ , and } \beta\ c_2 = 0.7639...$$

Thus

$$F_N = \beta\ c_N\ G_{N+1} < G_{N+1} \quad \text{for} \quad N \geq 2 .$$

Combined together the following bounds for F_N hold:

$$\frac{1}{G_{N+1}} < \frac{1}{F_N} < \frac{1}{G_N} \qquad \text{for} \quad N \geq 2 .$$

As expected the golden-section reduction $1 / G_N$ is inferior to the (optimal) Fibonacci reduction $1 / F_N$ (disregarding ε in (2.1o)). However, one additional function evaluation already makes the golden-section search superior to the Fibonacci search with N calls.

Comparing the four sequential techniques discussed so far we find:

1) Equally spaced search: $r_N = \frac{1}{(\sqrt{1.5})^N} \cong \frac{1}{(1.225)^N}$

2) Dichotomous search: $r_N \cong \frac{1}{(\sqrt{2})^N} \cong \frac{1}{(1.414)^N}$

3) Fibonacci search: $r_N \cong \frac{1}{F_N} \cong \frac{1.3817}{(1.618)^N}$

4) Golden-section search: $r_N = \beta^{N-1} \cong \frac{1.6180}{(1.618)^N}$

We conclude that for large N searches 3) and 4) are essentially more efficient than 1) or 2), however the golden-section search is only 1.17 times less efficient than the Fibonacci search, therefore the additional computational effort involved in 3) practically is not worth the 17% gain over the golden-section search.

2.3. Function approximation techniques

The searches described in the last paragraph can only be applied if the function $f(x)$ is unimodal which usually means convex. Their advantage then will be: They allow to bracket the minimum either with desired accuracy, then N will be fixed, or with a given number N of function evaluations, then accuracy is no longer freely available. In either case the particular structure of $f(x)$ is irrelevant.

Regardless of the fact that almost always the function $f(x)$ is not known in advance to be convex, the methods, which are based on function comparisons only, waste much information: they do not make use of the actual function value. In the following we describe a few methods which do use this information, leading to certain low order interpolating polynomials.

2.3.1. Quadratic interpolation

A parabola interpolating $f(x)$ at x_1, x_2, x_3 with function values f_1, f_2, f_3 can be expressed by Lagrange interpolation as

$$P(x) = \sum_{i=1}^{3} f_i \, L_i(x) \quad , \quad L_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)} \quad .$$

Explicitly:

$$P(x) = \frac{1}{(x_1-x_2)(x_1-x_3)(x_2-x_3)} \{ (x-x_2)(x-x_3)(x_2-x_3) \, f_1 + (x-x_1)(x-x_3)(x_3-x_1) \, f_2 + (x-x_1)(x-x_2)(x_1-x_2) \, f_3 \} .$$

From $P'(x) = o$ we obtain $\hat{x}$, which can be used as an approximation for the true minimizing position x^* :

$$\hat{x} = \frac{1}{2} \frac{(x_2^2-x_3^2) f_1 + (x_3^2-x_1^2) f_2 + (x_1^2-x_2^2) f_3}{(x_2-x_3) f_1 + (x_3-x_1) f_2 + (x_1-x_2) f_3} \quad . \tag{2.18}$$

For numerical reasons another representation may be more suitable. Without restriction we assume for the rest that $x_1 < x_2 < x_3$. Let $\hat{x} = x_2 + s$, then

$$s = \frac{1}{2} \frac{(x_2-x_3)^2 (f_2-f_1) + (x_1-x_2)^2 (f_3-f_2)}{(x_3-x_2) (f_2-f_1) + (x_1-x_2) (f_3-f_2)} \quad , \tag{2.19}$$

and s will be a small quantity whenever x_1, x_2, x_3 are already close to x^* . This expression reduces the effect of rounding errors.

Expression (2.19) can be simplified considerably for equidistant points. Let $x_1 = x_2 - \delta$, $x_3 = x_2 + \delta$, then

$$\hat{x} = x_2 + \frac{\delta}{2} \frac{(f_1-f_3)}{(f_1-2f_2+f_3)} \quad . \tag{2.2o}$$

For later purposes we derive the corresponding formula for the function values. We have

$$P(x) = \frac{1}{2\delta^2} \{(x-x_2)(x-x_3)f_1 - 2(x-x_1)(x-x_3)f_2 + (x-x_1)(x-x_2)f_3\} \quad .$$

This can also be expressed as a Taylor series expansion :

$$P(x) = f_2 + P'(x_2)\ (x-x_2) + \frac{a_2}{2}\ (x-x_2)^2 \quad ,$$

$$P'(x_2) = \frac{1}{2\delta}\ (f_3-f_1)$$

$a_2 = \frac{1}{\delta^2}\ (f_1-2f_2+f_3)$, which is positive if $\hat{x}$ is a minimum of P .

Using (2.2o) we obtain the desired result

$$P(\hat{x}) = f_2 - \frac{1}{8} \frac{(f_3-f_1)^2}{(f_1-2f_2+f_3)} \quad . \tag{2.21}$$

Another expression for $\hat{x}$ is convenient if the second derivative a_2 of the quadratic approximation and only two function values f_1, f_2 are known. Let the interpolating polynomial be

$$P(x) = a_o + a_1 x + \frac{a_2}{2} x^2 \quad .$$

$$P'(x) = o : \hat{x} = - \frac{a_1}{a_2} \quad .$$

With

$$f_i = a_o + a_1 x_i + \frac{a_2}{2} x_i^2 \quad , \qquad i = 1,2 \quad ,$$

we obtain

$$f_1 - f_2 = (x_1 - x_2)\ \{a_1 + \frac{a_2}{2}\ (x_1 + x_2)\}$$

and

$$\hat{x} = \frac{1}{2}\ (x_1 + x_2) - \frac{(f_1 - f_2)}{(x_1 - x_2)\ a_2} \quad . \tag{2.22}$$

In the following we discuss three strategies which use the quadratic interpolation in minimization codes.

a) The method of Davies, Swann and Campey (DSC) [6].

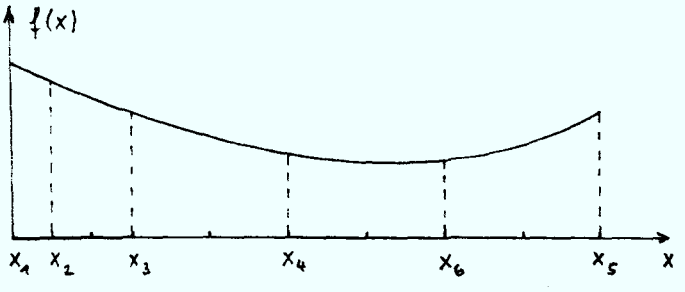


Fig. 2.1

Assume first that $f(x)$ has a negative slope at x_1 (later a direction having this property will be called a *descent direction*). For a given h the following sequence of points is generated:

$$x_{i+1} = x_i + 2^{i-1}h \,, \qquad i = 1,2,\ldots,$$

until for the first time $f(x_{\ell+1}) \geq f(x_\ell)$ with $\ell \geq 2$ (for h small enough we have either $f(x_2) < f(x_1)$, or the minimum is already located in $[x_1, x_2]$ within given accuracy). Then $f(x)$ is calculated at $x_{\ell+2} = \frac{1}{2}(x_\ell + x_{\ell+1})$, therefore the last four points $x_{\ell-1},\ldots, x_{\ell+2}$ are equally spaced (see e.g. Fig. 2.1, with $\ell = 4$). This allows to apply (2.2o), and the following outcomes are possible (for simplicity we assume $\ell = 4$).

By construction we always have $\quad f_3 > f_4 \,, \quad f_4 \leq f_5 \,.$

i) $f_4 \leq f_6$: x_5 is rejected and $\quad f_3 > f_4 \,, \; f_4 \leq f_6 \,,$

ii) $f_4 > f_6$: x_3 is rejected and $\quad f_4 > f_6 \,, \; f_6 < f_5 \,.$

In both cases i) and ii) a parabola can be constructed with its minimum at $\hat{x}$ according to (2.2o).

If $f(x)$ does not have a negative slope at x_1 try the negative axis as above, if this is not successful, x_1 will be accepted as the minimum position.

This procedure locates the minimum at $\hat{x}$, if the interpolation was successful, or at either x_4 or x_6, whichever yielded the smaller function value. A new sequence will be started with reduced step-length h, if the desired accuracy has not been achieved.

b) Powell's method |39|.

Calculate $f(x)$ at x_1 and $x_2 = x_1 + \delta$. We distinguish the two cases

i) $f_1 \geq f_2$. Then $x_3 = x_1 + 2\delta$,

ii) $f_1 < f_2$. Then $x_3 = x_1 - \delta$.

After calculating f_3 the interpolating parabola is constructed and $\hat{x}$ determined from (2.2o).

Now, as against to the DSC method, $\hat{x}$ will no longer be necessarily a minimum of the interpolating parabola. Also care should be taken to avoid too large extrapolation steps. For details of a combination of DSC with Powell's method we refer to Himmelblau |25|, pp. 46. Numerical results can also be found there.

c) Brent's method | 8 |.

This method combines golden-section search and parabolic interpolation in the same way as bisection and linear interpolation can be combined to find the zero of a function $f(x)$. In both cases the first method is safe but has only linear convergence, whereas the second method can fail but, if successful, reveals superlinear convergence. We omit the rather sophisticated details and refer to |8|, pp. 72. According to Brent this algorithm is not necessarily suitable for the use in an n-dimensional minimization procedure. An ALGOL 6o procedure is given in |8|, pp. 79.

2.3.2. Cubic interpolation

Cubic interpolation, using only function values, will in general not be used in algorithms for unconstrained minimization. If, however, first derivatives are also available, a method due to Davidon [14] is widely used. This method works as follows.

Let us assume that the new iteration of a minimization algorithm proceeds with a line search along a certain direction, say the x-axis, starting at x_o, where $f_o = f(x_o)$ and $g_o = f'(x_o)$ are known. Before the cubic interpolation can be applied, we need an estimate for the new point x_1. It is assumed that the decrease in function value, $\Delta f = f_o - f_1$, will be nearly the same as the one found during the previous iteration. With Δf as a known quantity, we estimate x_1 from the parabola

$$\bar{P}(x) = a(x - x_1)^2 + b ,$$

where a,b and x_1 follow from $\bar{P}(x_o) = f_o$, $\bar{P}'(x_o) = g_o$ and $\Delta f = f_o - \bar{P}(x_1)$. It follows that

$$x_1 = x_o - \frac{2\,\Delta f}{g_o} .$$

With f_1 and g_1 now available, the cubic interpolation polynomial can be determined. Without loss of generality we assume that $x_o = o$. Then

$$P(x) = f_o + g_o x + cx^2 + dx^3 ,$$

where c and d follow from $P(x_1) = f_1$, $P'(x_1) = g_1$, or

$$c = \frac{1}{x_1}(-3\delta - 2g_o - g_1)$$

$$d = \frac{1}{x_1^2}(2\delta + g_o + g_1) ,$$

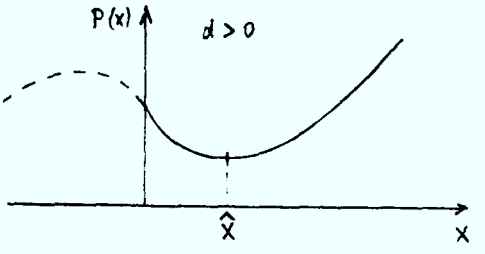
with $$\delta = \frac{1}{x_1}(f_o - f_1) .$$

From $P'(x) = o$ we obtain a prediction for the minimum

$$\hat{x} = \frac{1}{3d}\{-c \pm \sqrt{c^2 - 3g_o d}\} \qquad (2.23)$$

For the rest we assume that $g_o < o$, i.e. that the search line is a "descent direction", which in all applications will later be the case for the search directions under consideration. Three possibilities must be distinguished.

i) $d > o$: In this case $+\sqrt{\ }$ gives the minimum $\hat{x}$ of $P(x)$, irrespective of the sign of c .



ii) $d < o$: Real solutions only if $(c^2 - 3g_o d) > o$.

a) $c > o$: $+\sqrt{\ }$ corresponds to the minimum of $P(x)$.

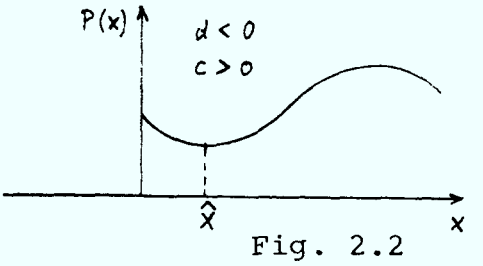


Fig. 2.2

b) $c < o$: $\{\ \} > o$ in (2.23), therefore $\hat{x} < o$.
This case must be excluded (see section 2.3.3. below).

It follows that only the upper sign of the square root in (2.23) is relevant. Next we derive Davidon's expression for $\hat{x}$.

$$\hat{x} = \frac{1}{3d}\left\{-c + \sqrt{c^2 - 3g_o d}\ \right\} = \frac{-g_o}{\sqrt{c^2 - 3g_o d} + c} .$$

With

$$c^2 - 3g_o d = \frac{1}{x_1^2}\left\{(3\delta + g_o + g_1)^2 - g_o g_1\right\}$$

and

$$z := (3\delta + g_o + g_1) , \quad w := \sqrt{z^2 - g_o g_1} :$$

$$\hat{x} = \frac{-g_o}{(w - z - g_o)}\, x_1 .$$

If x_1 is already a good approximation to the minimum x^* , we may reduce rounding errors by writing $\hat{x} = (1 - u)x_1$, where

$$u = \frac{(w - z)}{(w - z - g_o)} . \qquad (2.24a)$$

This representation, however, can numerically be disadvantageous, if namely $|g_o|$ is small against z , and z is assumed positive.

In this case we have

$$w \cong z - \frac{1}{2z}\, g_o g_1 \ ,$$

and therefore $(w-z)$ and $(w-z-g_o)$ are proportional to the small quantity g_o . Theoretically we obtain

$$u \cong \frac{g_1}{(2z + g_1)} \ ,$$

but numerically the presence of g_o in numerator and denominator may cause large rounding errors.

Davidon therefore suggested the representation

$$u = \frac{(w - z + g_1)}{(2w + g_1 - g_o)} \ , \qquad (2.24b)$$

which is easily verified to be equivalent to (2.24a) by equating both expressions. Now

$$(w - z + g_1) \cong g_1 \ , \quad (2w + g_1 - g_o) \cong (2z + g_1) \ ,$$

and the division by g_o is avoided.

We note, however, that for *negative z values* and g_o small we have $u \cong 1$, and the introduction of u only causes rounding errors. On the other hand, whenever z is negative, we have always $(w - z - g_o) > o$, and the original representation for $\hat{x}$ should be preferred.

The minimization codes working with gradients usually contain the following statements based on Davidon's formula:

```
Z = 3.* δ + g_o + g_1

W = SQRT (Z*Z - g_o* g_1)

Z = 1. - (W - Z + g_1) / (2.* W + g_1 - g_o)

x̂ = Z * x_1 .
```

This section only described how to obtain a prediction $\hat{x}$ for the minimum, when derivatives are available. An actual computer code must have a complete strategy, and a successful one will be given in the following section.

2.3.3. A line search strategy using Davidon's technique

This section describes a line search strategy which is used in the HARWELL subroutines VA08AD (Fletcher-Reeves method, |17|) and VA09AD (Fletcher's switching algorithm |18|) both requiring analytical derivatives. In subroutine VA13AD (Powell's version of BFGS |4o|) a similar but more refined strategy is implemented.

In the following we assume as before that $g_o < o$, and that x_1, f_1, g_1 are already known. The two cases $f_1 \geq f_o$ and $f_1 < f_o$ will be considered separately.

i) $f_1 \geq f_o$.

In this case a local minimum must exist in (x_o, x_1) , and eq. (2.23) has real roots, no matter what sign g_1 will have. The two possible outcomes are illustrated in Fig. 2.3.

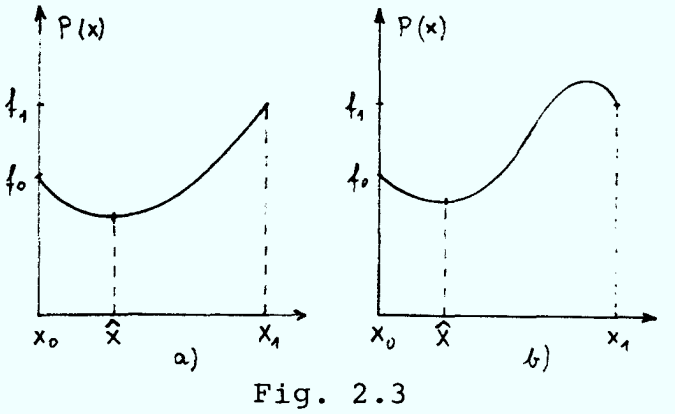


Fig. 2.3

In case i) the cubic interpolation can be applied without any restrictions. The predicted minimum $\hat{x}$ is used as the new value for x_1 , and, depending on $f(\hat{x}) \geq f_o$ or $f(\hat{x}) < f_o$, the algorithm continues with step i) or ii) respectively.

ii) $f_1 < f_o$.

a) $|g_1| \leq \delta_o |g_o|$, $\quad o < \delta_o < 1$.

This is the termination criterion for the line search. Especially for $g_1 = o$ the exact minimum x^* is found. x_1 is accepted and a new search direction is chosen. VA08AD and VA09AD use $\delta_o = o.9$.

b) $g_1 > \delta_o |g_o|$.

This case is completely analogous to the case i) treated above (except that g_1 is known to be positive and therefore Fig. 2.3b is not possible).

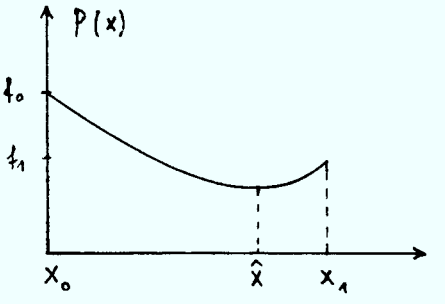


Fig. 2.4

c) $g_1 < -\delta_o |g_o|$.

Now a minimum may or may not exist in (x_o, x_1). This case also includes case ii) b) from section 2.3.2. (see Fig. 2.5c), which had to be excluded from the cubic interpolation. Fig. 2.5 illustrates the three possible types.

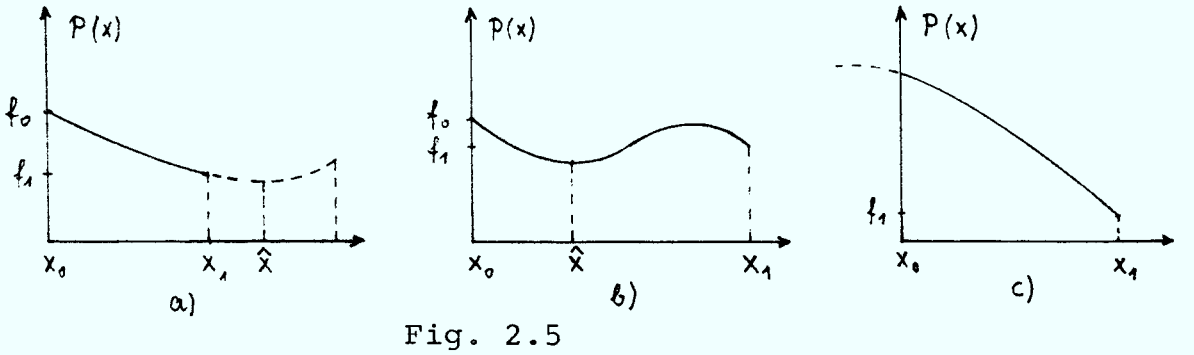


Fig. 2.5

As $f_1 < f_o$ *and* $g_1 < o$, the "direction" seems most promising and an *extrapolation step* will be tried as follows.

$|g_1| \geq |g_o|$: A steep down-hill direction, a large step may be successful, try $x_1' = x_o + 1o(x_1 - x_o)$.

$|g_1| < |g_o|$: Because of a) this can only happen if $\delta_o |g_o| < |g_1| < |g_o|$, which is a small range for $\delta_o = o.9$. The extrapolation step gives

$$x_1' = x_o + \frac{g_o}{(g_o - g_1)} (x_1 - x_o) \quad . \qquad (2.25)$$

(See the derivation below). To avoid too large steps, an upper limit of $1o(x_1 - x_o)$ will be prescribed.

After the extrapolation step,the search continues at i) or ii) with

$$x_o := x_1 \ , \quad f_o := f_1 \ , \quad g_o := g_1 < o \ ,$$

$$x_1 := x_1' \ , \quad f_1 := f(x_1') \ , \quad g_1 := g(x_1') \ .$$

This is again a "descent direction" because of $g_o^{(new)} < o$.

Eq. (2.25) is based on the assumption that the gradient $g(x)$ can be approximated linearly by

$$\bar{g}(x) = g_o + \frac{(g_1 - g_o)}{(x_1 - x_o)} (x - x_o) \ .$$

The predicted minimum is then determined from $\bar{g}(\hat{x}) = o$ leading to (2.25) with $x_1' = \hat{x}$. As used in the context of (2.25), this corresponds indeed to an extrapolation, because

$$\frac{g_o}{(g_o - g_1)} = \frac{|g_o|}{|g_o| - |g_1|} > 1 \ .$$

We observe that the condition $|g_1| < |g_o|$ is always realized in Fig. 2.5a). Here the assumption of a linear gradient is acceptable and a good one. However, $|g_1| < |g_o|$ can also be realized in the case illustrated in Fig. 2.5b), and a linear gradient is no longer meaningful.

When this strategy was implemented in a minimization algorithm for the first time, it introduced the basically new concept of *approximate line search*, expressed by the choice $\delta_o > o$. We do not give here any details about conditions, which are necessary in order to have convergence, as this is a difficult subject and would be out of the scope of this short example for a line search strategy. It may have become clear, however, that the line search will be a major part of any minimization code and should preferably be treated as a modular part of the code.

3. Direct search methods

From now on we assume $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}$ is the function to be minimized. In this chapter we shall only briefly discuss the so-called direct search methods, the powerful minimization methods will later be presented in great detail.

Direct search or stepping methods are characterized by the fact that they make only little use of the information gained about the objective function during the minimization process. Moreover, line searches are not performed in general, one reason being that the concept of a search direction is not clearly defined. As a consequence, lack of convergence or an extremely slow rate of convergence will frequently be encountered, especially if the dimension of a problem is high, where four dimensions may already be considered high. The methods tend to run into difficulties at "sharp corners", "curving valleys" or "steep ridges", which may already occur during the early stages of the minimization. The geographical rather than mathematical terms used here shall indicate that these "hill-climbing" methods were originally designed intuitively, guided by two-dimensional problems.

Yet some undeniable advantages must be stated:

i) The algorithms are very simple in structure.
ii) The algorithms are very robust, e.g. no singular matrices or difficulties with line searches can occur.
iii) No gradients are required, the objective function may even have discontinuous derivatives.
iv) Simple constraints like upper and lower bounds can easily be handled.
v) The algorithms can be implemented in small computers.

In other words, these methods can usually be programmed easily, the storage requirements are negligible, no line search strategies nor complicated matrix updating schemes are necessary thus making the algorithms extremely fool-proof.

A classification roughly allows two groups to be distinguished:

algorithms using a deterministic search, and so-called random search methods. The three classical methods are:

i) The method of Hooke and Jeeves |26|.
ii) The simplex method of Nelder and Mead |34|.
iii) Rosenbrock's method |44|.

Almost all other direct search methods are modifications of these basic types. We list some of them. Box |7| generalized ii) in order to deal with constraints (the "complex method"), Spendley |49| enabled to obtain the covariance matrix when ii) is used to solve nonlinear least-squares problems, Davies, Swann and Campey |6| improved iii) by adding an accurate line search, and Burhardt |11| introduced an adaptive search technique in iii). Most random search techniques are modifications of i) (see e.g. Schrack and Borowski |46| for a numerical comparison of three versions of these methods), further we mention the randomized pattern search by Lawrence and Steiglitz |30| and an adaptive random search by Beltrami and Indusi |5|.

In this chapter only methods ii) and iii) will be described in more detail.

3.1. The simplex method

The simplex method was originally designed by Spendley et al |50| and later refined by Nelder and Mead |34|. The function $f(x)$ is calculated at the points of a simplex, i.e. at $(n + 1)$ points which span a n-dimensional volume element. We assume for simplicity that the function values are ordered:

$$f_o < f_1 < \ldots < f_n \quad \text{at} \quad x_o, x_1, \ldots, x_n \quad \text{respectively.}$$

The worst point x_n is discarded and replaced by a new one according to the strategy given below.

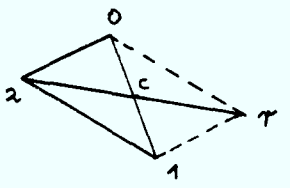
The mean of the remaining n points is given by

$$x_c = \frac{1}{n} \sum_{i=o}^{n-1} x_i \ .$$

x_n is reflected with respect to x_c yielding the point

$$x_r = x_c + \alpha(x_c - x_n) \ ,$$

where the *reflection coefficient* α will usually be set equal to one.



Reflection

(for $n=2$, $\alpha=1$)

Three cases are distinguished:

1) $f_o \leq f_r \leq f_{n-1}$,

2) $f_r < f_o$, (x_r would be a new best point)

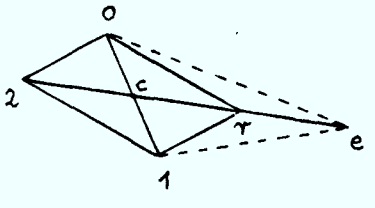
3) $f_r > f_{n-1}$. (x_r would be a new worst point)

Case 1) x_r is accepted as new point, the function values are reordered and the next iteration is started.

Case 2) The direction $x_r - x_n$ is promising, try the expansion

$$x_e = x_c + \gamma(x_r - x_c) \ ,$$

γ is the *expansion coefficient*, usually $\gamma = 2$.
If $f_e < f_r$, the expansion was successful, set $x_o := x_e$ for the next iteration, otherwise $x_o := x_r$.



Expansion

(for $n=2$, $\gamma=2$)

We remark that in the original treatment |34| of the method the function value f_e is compared with f_o , and not with f_r , although it is known that $f_r < f_o$. Then x_e is accepted as a new point if $f_e < f_o$, even if $f_e > f_r$, which seems unreasonable and is obviously an error. (It can also be found in Murray |33|, p. 27, Himmelblau |25|, p. 151, Box et al |6|, p. 22, Kowalik and Osborne |28|, p. 26 , whereas in Davies |15|, p. 9o, the correct strategy is given.)

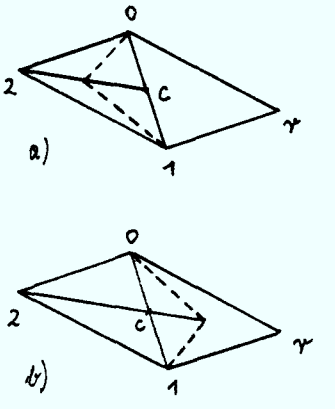
Case 3) A contraction of the line $x_r - x_n$ is tried.

3a) $f_r \geq f_n$: $x_k = x_c + \beta(x_n - x_c)$

3b) $f_r < f_n$: $x_k = x_c + \beta(x_r - x_c)$

β is the *contraction coefficient*, usually $\beta = 0.5$.

If no improvement can be obtained, the size of the simplex must be reduced.



Contraction

(for $n=2$, $\beta=0.5$)

Like the method itself, the termination criterion is very simple. Let $\bar{f}$ denote the mean value of all function values averaged over the simplex

$$\bar{f} = \frac{1}{(n+1)} \sum_{i=0}^{n} f_i ,$$

then the variance

$$\sigma^2 = \frac{1}{n} \sum_{i=0}^{n} (f_i - \bar{f})^2$$

is used to terminate the search whenever $\sigma < \varepsilon$ for some prescribed ε .

Recently Olsson and Nelson |35| gave numerical results for the simplex method minimizing 6 low-dimensional problems ($n \leq 4$) including constraints and discontinuous first derivatives. Their results seem quite satisfactory, which can to some extent be attributed a) to the small number of variables, and b) to the fact that the performance of more powerful methods has not been demonstrated for these examples. When both restrictions are absent, a rather poor performance may result (see e.g. Amadori et al |2|).

Fig. 3.1 (Fig. 5.2 of |2|) exhibits a typical danger for this method (as objective function a nonlinear least-squares problem of moderate degree of difficulty with $n=9$ is selected. The circles represent the simplex method). The simplex may collapse

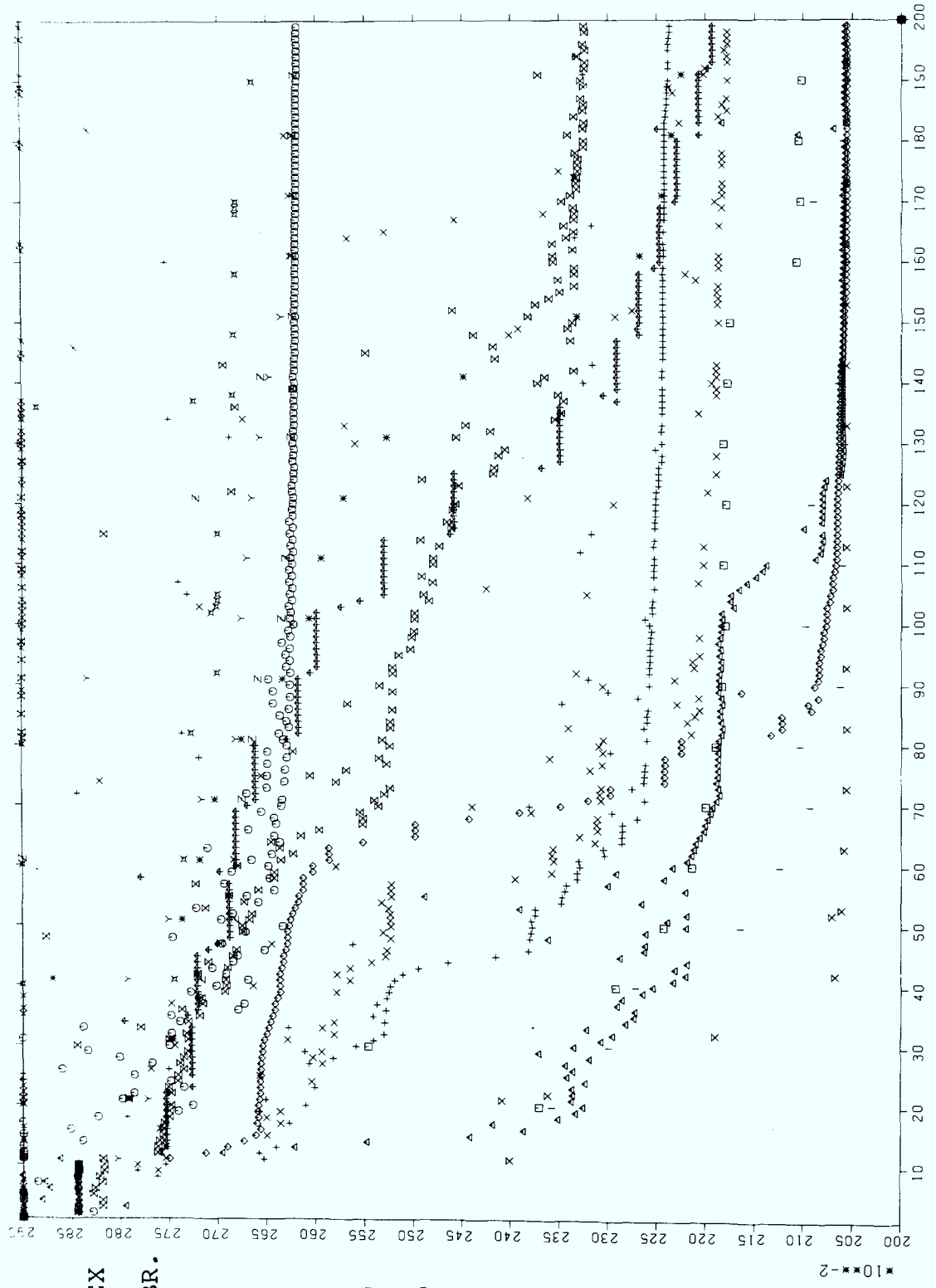


Fig. 3: Plot of $\log_{10} f$ against EFE for a nonlinear least squares problem

and, as a consequence, the search covers no longer the n-dimensional space, thus it becomes impossible to reduce the function value. To overcome this difficulty at least to a certain degree, Box [7] introduced the "complex", which has more than (n + 1) points, typically 2n points.

It should be stressed however, that the examples of Olsson and Nelson show clearly a good performance for the functions with discontinuous derivatives, where most other algorithms would fail. It would be desirable that a general subroutine library contains a good implementation of the simplex method, but the user should be aware of its weaknesses.

3.2. Rosenbrock's method

The method of Rosenbrock - like the simplex method - makes only use of function comparisons. In a certain sense it can be regarded as a predecessor of Powell's method (see chapter 4), as it also uses the overall progress made during one iteration in order to build up the new set of search directions.

Iteration k is defined as follows. Given a set of orthonormal vectors $s_i^{(k)}$, $i = 1,\ldots,n$; for $k = 1$ these are parallel to the coordinate axes. Let $\lambda_i^{(k)}$ be the initial step-length for direction $s_i^{(k)}$ at the beginning of iteration k, and $\bar{x}_o^{(k)}$ the starting point. Then iteration k proceeds as follows.

Set $x_o := \bar{x}_o^{(k)}$, and for $i = 1,\ldots,n$ update x_o successively according to the scheme

$$x' = x_o + \lambda_i^{(k)} s_i^{(k)} ,$$

i) $f(x') \leq f(x_o)$ (success) : $x_o := x_o + \lambda_i^{(k)} s_i^{(k)}$,
$\lambda_i^{(k)} := \alpha \lambda_i^{(k)}$, $(\alpha = 3)$

ii) $f(x') > f(x_o)$ (failure) : x_o remains unchanged,
$\lambda_i^{(k)} := \beta \lambda_i^{(k)}$. $(\beta = -0.5)$

After this cycle has been completed, another cycle is started with the same directions but $\lambda_i^{(k)}$ as modified before depending on success (S) or failure (F). This process is continued, and for each direction a sequence of S and F is recorded. The process is terminated, when for the first time *all* sequences contain a S-F pair at least once, i.e. along each direction a failure must have followed a success for two consecutive cycles. We show that this in fact is possible.

Since $\alpha > 1$, there cannot be arbitrarily many successes in succession for one sequence. On the other hand, at least one success must occur in each direction after a certain number of failures, because the step-length for successive failures is always reduced and reversed, so that for small enough steps at least *equal* function values will be encountered, which is registered as success.

If $\bar{x}_o^{(k+1)}$ denotes the final value of x_o, then $\bar{x}_o^{(k+1)} - \bar{x}_o^{(k)}$ is the direction of total progress. Let γ_i denote the total step made along direction $s_i^{(k)}$. Define the vectors

$$
\begin{aligned}
u_1 &= \gamma_1 s_1^{(k)} + \gamma_2 s_2^{(k)} + \ldots + \gamma_n s_n^{(k)} \\
u_2 &= \qquad\quad \gamma_2 s_2^{(k)} + \ldots + \gamma_n s_n^{(k)} \\
&\vdots \qquad\qquad\qquad\qquad \vdots \\
u_n &= \qquad\qquad\qquad\quad \gamma_n s_n^{(k)}
\end{aligned}
$$

Then $u_1 = (\bar{x}_o^{(k+1)} - \bar{x}_o^{(k)})$ by definition of the γ_i. If all $\gamma_i \neq o$, the u_i are linearly independent and the Gram-Schmidt procedure can be applied to define the new set of directions.

$$
s_1^{(k+1)} = \frac{u_1}{\|u_1\|},
$$

$$
s_i^{(k+1)} = \frac{v_i}{\|v_i\|}, \quad \text{with} \quad v_i = u_i - \sum_{j=1}^{i-1} (u_i^T s_j^{(k+1)})\, s_j^{(k+1)}
$$

$$
i = 2,\ldots,n
$$

Since $s_1^{(k+1)}$ points along the direction of overall progress, one can hope that the first direction is eventually aligned along the principal axis with greatest elongation of the contours of $f(x)$, where we assume that $f(x)$ can be approximated by a quadratic function close to the minimum.

It remains to show that $\gamma_i \neq o$. Let us consider the following example for a sequence (assume $\lambda_i^{(k)} = 1$ initially) :

S	S	F	F	F	S	F	S	S	S	F	S
1	3	3^2	$-\frac{3^2}{2}$	$\frac{3^2}{2^2}$	$-\frac{3^2}{2^3}$	$-\frac{3^3}{2^3}$	$\frac{3^3}{2^4}$	$\frac{3^4}{2^4}$	$\frac{3^5}{2^4}$	$\frac{3^6}{2^4}$	$-\frac{3^6}{2^5}$

The lower line gives the actual value of λ which is the result of the outcome of the step before (e.g. the first "3" corresponds to the first S).

The total step for this example would be

$$\gamma = 1 + 3 - \frac{3^2}{2^3} + \frac{3^3}{2^4} + \frac{3^4}{2^4} + \frac{3^5}{2^4} - \frac{3^6}{2^5} .$$

From this expression we easily obtain the general form. Let m be the total number of successes in a given sequence, and let ν label these from left to right, $\nu = o,\ldots,m-1$. Let r_ν denote the number of failures appearing up to ν. For the example above, $m = 7$ and

$$r_o = r_1 = o , \quad r_2 = 3 , \quad r_3 = r_4 = r_5 = 4 , \quad r_6 = 5 .$$

Consider the polynomial

$$P_n(x) = \sum_{\nu=o}^{n} (-\beta)^{r_\nu} x^\nu , \tag{3.1}$$

then it follows that $\gamma = P_{m-1}(\alpha)$. If we write

$$P_n(x) = (-\beta)^{r_n} \sum_{\nu=o}^{n} a_\nu x^\nu , \qquad a_\nu = (-\frac{1}{\beta})^{r_n - r_\nu} ,$$

then all coefficients a_ν are powers of 2 for $\beta = -0.5$, the constant term a_o is not divisible by $\alpha = 3$, therefore α cannot be a zero of $P_{m-1}(x)$. Or in other words :
The expression

$$(3a_1 + 3^2a_2 + \ldots + 3^na_n)$$

is an integer divisible by 3, adding a_o cannot annihilate this expression.

If the sequence starts with ℓ failures, then the first term in γ will be $(-\beta)^\ell$ which can be factored out and the arguments above still hold. Finally, since one success is at least recorded, we always have $\gamma \neq o$.

Remark: In the literature generally the conditions $\alpha > 1$ and $-1 < \beta < o$ can be found. Rosenbrock |44| recommended $\alpha = 3$ and $\beta = -0.5$. From the discussion above we conclude that α and β cannot be chosen quite arbitrarily, instead care must be taken that α does not become a zero of $P_n(x)$ from (3.1). If we select e.g. $\alpha = 2$ and $\beta = -0.5$, a sequence SFSF, for instance, would lead to $\gamma = 1 - 1 = o$, and the search directions $s_i^{(k+1)}$ would no longer be linearly independent.

4. Conjugate direction methods

In this chapter an important class of minimization algorithms will be presented which does not use gradients, even numerical derivatives are not required. However, these algorithms are different from the direct search methods discussed in chapter 3, because the minimum along given directions must be found. The objective function is assumed to possess continuous first and second derivatives.

As opposed to the direct search methods these methods can be very powerful and comparable in their efficiency - at least for not too high dimensions, typically $n \leq 20$ - to variable metric methods which will be the major subject of this report. The most prominent representative of the class of derivative-free methods is Powell's 1964 method [39], which is also available as an excellent subroutine (e.g. VA04AD). This method will therefore be described in detail. Some modifications of Powell's original method will also be discussed.

Quadratic forms

$$f(x) = \frac{1}{2} x^T G x + b^T x + c \ , \quad G \in \mathbb{R}^{(n,n)} \ , \ b \in \mathbb{R}^n \ , \ c \in \mathbb{R} \ ,$$

with positive definite matrix G play a fundamental role in the design of minimization algorithms, since in general most objective functions will behave like a quadratic function near a local minimum. An algorithm minimizing a quadratic function in a small number of steps is also supposed to exhibit fast ultimate convergence when applied to a non-quadratic function. The problem of minimizing a quadratic function *in a finite number of steps* leads in a most natural way to the concept of *conjugate directions*.

4.1. Conjugate directions

Definition: A set of nonvanishing vectors p_i , $i = 0(1)\ m - 1$ with $m \leq n$, is said to be *conjugate with respect to a positive definite matrix* A if

$$p_i^T A p_j = 0 \ , \quad i \neq j \ . \tag{4.1}$$

The vectors p_i are called *conjugate directions*. Some important properties will be given below.

i) Conjugate directions are linearly independent.

Let $u \in \mathbb{R}^m$ be a vector in the space spanned by p_i, $u = \sum_{i=o}^{m-1} c_i p_i$. Then from (4.1)

$$p_j^T A u = c_j p_j^T A p_j , \quad j = o\ (1)\ m - 1 ,$$

and from $u = o$ we have $c_j = o$, since A is positive definite. (In particular, the set of p_i is complete for $m = n$).

ii) Eigenvectors form a special complete set of conjugate directions:

$$p_i^T A p_j = \lambda_j p_i^T p_j = o , \quad i \neq j ,$$

since the eigenvectors p_i of a positive definite matrix are orthogonal (note that positive definiteness implies the symmetry of A).

For the following we need the definition of a perfect line search.

Definition: A *perfect* (or exact) *line search* finds a local minimum $\lambda = \alpha$ of

$$h(\lambda) = f(x + \lambda p) .$$

$h'(\lambda)$ is called *directional derivative* of f along the direction p.

Let g be the gradient of f, then

$$h'(\alpha) = p^T g(x + \alpha p) = o , \qquad (4.2)$$

i.e. the gradient taken at $\lambda = \alpha$ is orthogonal to p.

iii) (The fundamental property of conjugate directions.) The minimum of a quadratic form

$$f(x) = \frac{1}{2} x^T G x + b^T x + c \qquad (4.3)$$

with positive definite G will be found in at most n perfect line searches, if these are taken along n directions p_i conjugate with respect to G. Each direction is used once, the order in which the p_i are chosen is irrelevant.

Proof a) (Fletcher-Reeves |19|). Let x_o be an arbitrary starting point, and

$$x_{i+1} = x_i + \alpha_i p_i , \qquad o \leq i \leq n - 1 \qquad (4.4)$$

where the α_i are determined from (4.2) with $x = x_i$ and $p = p_i$.

With $g(x) = G x + b = o$ at the minimum x^*, we have $b = -G x^*$, and

$$g_i = G (x_i - x^*) , \qquad o \leq i \leq n . \qquad (4.5)$$

From (4.4) :

$$x_n = x_i + \sum_{j=i}^{n-1} \alpha_j p_j , \qquad o \leq i \leq n - 1 ,$$

and if we combine this equation for $i + 1$ with (4.5), we obtain

$$g_n = G (x_{i+1} - x^*) + \sum_{j=i+1}^{n-1} \alpha_j G p_j$$

$$= g_{i+1} + \sum_{j=i+1}^{n-1} \alpha_j G p_j , \qquad o \leq i \leq n - 2 .$$

Using the conjugacy property and the perfect line search property (4.2) which can be written as

$$g_{i+1}^T p_i = o , \qquad o \leq i \leq n - 1 , \qquad (4.6)$$

one finally obtains

$$g_n^T p_i = g_{i+1}^T p_i + \sum_{j=i+1}^{n-1} \alpha_j p_j^T G p_i = o \ , \quad o \leq i \leq n-2 \ ,$$

and for $i = n-1$: $g_n^T p_{n-1} = o$ from (4.6).

Therefore g_n is orthogonal to the complete set of vectors p_i and must vanish identically. From (4.5)

$$g_n = G (x_n - x^*) = o \ ,$$

which proves that $x_n = x^*$, i.e. the minimum is located after at most n line searches. If some of the α_i turn out to be zero, the minimum is already located earlier.

Proof b) (Powell |39|). Since the set of p_i is complete, we may represent any vector x as

$$x = x_o + \sum_{i=o}^{n-1} c_i p_i \ .$$

This inserted into (4.3), and using $p_i^T G p_j = o$ for $i \neq j$, leads to

$$f(x) = f\left(x_o + \sum_{i=o}^{n-1} c_i p_i\right) = f(x_o) + \sum_{i=o}^{n-1} h_i(c_i) \ ,$$

with

$$h_i(c) = \frac{1}{2} c^2 p_i^T G p_i + c p_i^T (G x_o + b) \ . \qquad (4.7)$$

Therefore, in order to find the minimum of the quadratic function $f(x)$, starting at x_o, is equivalent to the problem of finding the minimum with respect to the arbitrary constants $c_o, \ldots, c_{n-1}$. The minimization with respect to a particular c_i is equivalent to a perfect line search along p_i. Since the c_i are not correlated to each other, i.e. products $c_i c_k$ with $i \neq k$ do not occur, there are at most n perfect line searches, and it is irrelevant in which order these minimizations take place.

This proof allows in particular to interpret iii) in a different way, which will be important in later applications:

iii') Given a set of $k \leq n$ mutually conjugate directions $p_o,\ldots,p_{k-1}$ and an initial point x_o. Then perfect line searches along $p_o,\ldots,p_{k-1}$, starting at x_o, allow to locate the minimum of f, if x is restricted to the subspace spanned by $p_o,\ldots,p_{k-1}$ and which contains x_o. The order in which the p_i are chosen is immaterial.

As a special case of (4.5) we obtain for $i = o$ the Newton method

$$x^* = x_o - G^{-1} g_o .$$

If the inverse of the matrix G and the gradient $g(x_o)$ are known, a quadratic function is minimized in one (Newton) step. The inverse G^{-1} can be constructed by means of the conjugate directions in the following way.

iv) Let $p_o,\ldots, p_{n-1}$ be a complete set of conjugate directions with respect to G. Then

$$G^{-1} = \sum_{i=o}^{n-1} \frac{p_i\, p_i^T}{p_i^T\, G\, p_i} . \qquad (4.8)$$

Proof (Kowalik-Osborne |28|). Let

$$A = \sum_{i=o}^{n-1} \frac{p_i\, p_i^T}{p_i^T\, G\, p_i} ,$$

then

$$AG\, p_j = p_j , \qquad j = o\ (1)\ n - 1 .$$

The conjugate directions p_j are therefore eigenvectors of AG with eigenvalues $\lambda_j = 1$. With $S = (p_o,\ldots,p_{n-1})$ and $D = \mathrm{diag}(\lambda_j) = I$ we can represent AG as

$$AG = SDS^{-1} = I ,$$

which proves $A = G^{-1}$.

Property iv) allows to give a third proof of iii).

Proof c) (Kowalik-Osborne [28]). From

$$x_{i+1} = x_o + \sum_{j=o}^{i} \alpha_j p_j$$

and the perfect line search condition

$$p_i^T g(x_{i+1}) = p_i^T (G x_{i+1} + b) = o ,$$

we obtain

$$p_i^T (G x_o + \alpha_i G p_i + b) = o ,$$

thus

$$\alpha_i = - \frac{1}{p_i^T G p_i} p_i^T (G x_o + b) .$$

Finally

$$x_n = x_o + \sum_{j=o}^{n-1} \alpha_j p_j = x_o - \sum_{j=o}^{n-1} \frac{p_j p_j^T}{p_j^T G p_j} (G x_o + b)$$

$$= x_o - G^{-1} (G x_o + b) = -G^{-1} b ,$$

where property iv) has been used. With $b = -G x^*$ the desired result follows.

Until now we only presented some important properties of conjugate directions and were not concerned with the question, how to generate them. The next property will describe how to construct conjugate directions *without knowledge of gradients*.

v) *Parallel subspace property*: Given a quadratic function and two parallel lines with direction p through x_o, x_1, respectively. If perfect line searches are performed along p yielding the minima y_o, y_1, respectively, the direction $y_1 - y_o$ is conjugate to p.

Proof (Fletcher |2o|). With $f(x)$ from (4.3) and

$$h_j(\lambda) = f(x_j + \lambda p), \qquad j = o,1 ,$$
$$h_j'(\lambda) = p^T g(x_j + \lambda p),$$

the perfect line search gives $p^T g(y_j) = o$, $j = o,1$, or explicitly

$$p^T (Gy_j + b) = o , \qquad j = o,1 .$$

Subtraction leads to the desired result $p^T G(y_1 - y_o) = o$.

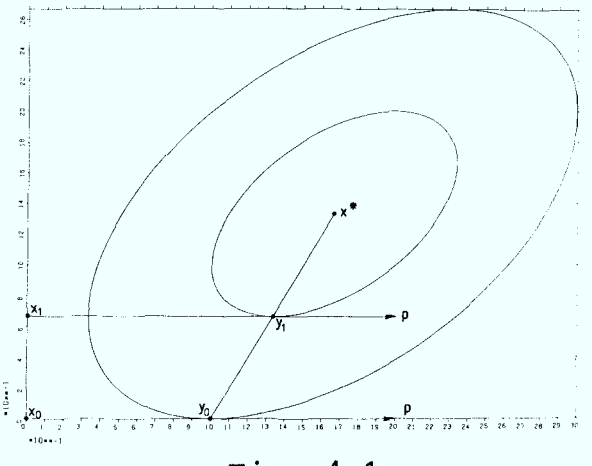


Fig. 4.1

Fig. 4.1 illustrates this property in two dimensions. From property iii) we know that after $n=2$ exact line searches along conjugate directions the minimum must be found. In Fig. 4.1 the directions P and $(y_1 - y_o)$ are conjugate, thus *three* exact line searches (one is necessary to generate the second conjugate direction) lead to the minimum.

Next we consider the generalization of v) for k-dimensional subspaces.

vi) Given k linearly independent directions $d_o,\ldots,d_{k-1}$, with $k < n$, and two points $x_j \in \mathbb{R}^n$, $j = o,1$, such that the two k-dimensional spaces

$$S_j = \{ z \mid z = x_j + \sum_{i=o}^{k-1} c_i d_i \} , \quad j = o,1 , \; c_i \in \mathbb{R} ,$$

are different. Then the direction $y_1 - y_o$, where the y_j are the two minima of the quadratic function $f(z)$ with $z \in S_j$ respectively, is conjugate to all d_i , or

$$d_i^T G (y_1 - y_o) = o , \qquad i = o(1) \; k - 1 .$$

Proof. Let y_j be the minimum of $f(x)$ with $x \in S_j$. Then, by definition, a perfect line search along any line d_i which passes through y_j must find the point y_j, as this is the unique minimum in S_j. Therefore the directional derivative taken at y_j along d_i must vanish, or explicitly

$$d_i^T (Gy_j + b) = o , \quad j = o,1 , \quad i = o(1)k-1 ,$$

and after subtraction $d_i^T G (y_1 - y_o) = o$, $i = o(1)k-1$.

Remark. Property vi) holds for *any* set of d_i. However, only for *conjugate* d_i one can locate the minimum in S_j after k perfect line searches starting from x_j (or any other point $x_j' \in S_j$). On the other hand, if the d_i are not conjugate, the point reached after k searches will depend on the starting position $x_j' \in S_j$, and also on the order in which the line searches along the d_i are performed.

Properties v) and vi) form the basis for all minimization algorithms using conjugate directions without calculating derivatives. An additional property will be needed when deriving Powell's second procedure, see section 4.3.2.

vii) Let the linearly independent directions $d_o, \ldots d_{n-1}$ be scaled such that

$$d_i^T G d_i = 1 , \qquad i = o(1)\ n - 1 . \qquad (4.9)$$

Then the determinant of the matrix $A = (d_o, \ldots, d_{n-1})$ takes its maximum value, if the d_i are mutually conjugate.

Proof. Given a set of conjugate directions $p_o, \ldots p_{n-1}$, scaled as in (4.9). Then a nonsingular transformation U exists relating d_i and p_i:

$$d_i = \sum_j U_{ij} p_j , \quad i = o(1)\ n - 1 ,$$

which with $B = (p_o,\ldots,p_{n-1})$ can also be written as $A^T = UB^T$. With

$$d_i^T G d_j = \sum_k \sum_\ell U_{ik} U_{j\ell} p_k^T G p_\ell = \sum_k U_{ik} U_{jk} \qquad (4.1o)$$

we find in particular for $i = j$, using (4.9) ,

$$d_i^T G d_i = \sum_k (U_{ik})^2 = 1 , \qquad i = o(1)\ n - 1 .$$

Applying the Hadamard inequality to U therefore yields

$$(\det U)^2 \leq \prod_i \sum_k (U_{ik})^2 = 1 .$$

Now $|\det U|$ is the volume of the polyeder spanned by the columns of U , and its maximum value is achieved when all columns are orthogonal. Thus U must be orthogonal, the set of d_i then is conjugate which follows from (4.1o), and the determinant of A assumes its maximum possible value

$$|\det A| = |\det (UB^T)| = |\det B| .$$

4.2. The method of Smith

The first derivative-free method based on conjugate directions was suggested by Smith |48|. Initially a set of n linearly independent vectors $d_o,\ldots d_{n-1}$, $(n - 1)$ positive constants $\beta_1,\ldots,\beta_{n-1}$, and a starting approximation x_o are given. For a quadratic function the Smith algorithm then works as follows.

1) Set $p_o = d_o$ and find the minimum x_1 of $f(x)$ along p_o , starting at x_o .
2) Displace x_1 to $u = x_1 + \beta_1 d_1$, and find the minimum z of $f(x)$ along p_o , starting at u . Set $p_1 = z - x_1$ which from property v) is conjugate to p_o. Then find the minimum x_2 along p_1 , starting at z .

3) For $i = 2(1)n-1$, the point x_i is displaced to $u = x_i + \beta_i d_i$, and the point z is determined by successive perfect line searches along $p_o,\ldots,p_{i-1}$, starting at u . The new direction p_i is given by $p_i = z - x_i$, and the new point x_{i+1} is the minimum along p_i starting at z .

By an inductive argument it is obvious, that the directions $p_o,\ldots,p_{i-1}$ are conjugate. For $i=2$ this follows from step 2). Assume that $p_o,\ldots,p_{i-1}$ are conjugate. The new direction p_i is then conjugate to $p_o,\ldots,p_{i-1}$ according to property vi).

In the final iteration, z is found by minimizing along $p_o,\ldots,p_{n-2}$ starting at $u = x_{n-1} + \beta_{n-1}d_{n-1}$, and x_n by minimizing along p_{n-1} starting at z . Therefore x_n is found after n consecutive perfect line searches along conjugate directions and thus coincides with the minimum.

The total number of line searches follows from step 3), if we consider the move $x_i \rightarrow x_{i+1}$ which requires (i+1) searches. The total number to obtain x_n is $\frac{1}{2}\,n(n+1)$. Then p_o has been explored n times, p_1 $(n-1)$ times ,..., and p_{n-1} just once. For $n=2$ the Smith procedure, as given by steps 1) and 2), is illustrated in Fig. 4.1, if the notation is adequately changed. In fact three line searches are required.

When applying the Smith procedure iteratively to non-quadratic functions great difficulties have been encountered, which can be attributed to the unequal use of conjugate directions mentioned above (see Fletcher |2o|, p. 77). Also an appropriate choice of the β_i can become a significant factor.

In an attempt to improve Smith's method, Fletcher |21| suggested to replace the arbitrary steps β_i by perfect line searches along d_i , and to change cyclicly the initial set of directions $d_i^{(k)}$, $k = o,1,...,$ for the new iteration (k+1). The first suggestion yields optimal β_i , but can lead to linearly dependent directions

(Sutti |52|), e.g. if x_1 already coincides with the minimum along d_1 resulting in $\beta_1 = o$. The second suggestion saves one line search per iteration : After the first iteration has been completed, set $d_o^{(1)} := p_{n-1}^{(o)}$, $d_1^{(1)} := p_o^{(o)}$,..., $d_{n-1}^{(1)} := p_{n-2}^{(o)}$ and $x_o^{(1)} := x_n^{(o)}$ .As a result step 1) is no longer required, because $x_n^{(o)}$ is the minimum along $p_{n-1}^{(o)}$. The total number of line searches per iteration for Fletcher's modification of Smith's method therefore is

$$\frac{1}{2}\, n(n+1) + (n-1) - 1 = \frac{1}{2}\,(n-1)\,(n+4)\ ,$$

except for the first iteration which requires one further line search.

4.3. The method of Powell

Powell's method succeeded to overcome the difficulties arising with Smith's method, and has become one of the fundamental minimization algorithms. It is still one of the best derivative-free algorithms, although dating back to 1964. The basic concept will be described in the following section.

4.3.1. Powell's first procedure

Initially a set of n linearly independent directions $d_o^{(o)}$,..., $d_{n-1}^{(o)}$ - usually the coordinate directions - and a starting point $x_o^{(o)}$ are given. Then the first iteration proceeds as follows.

1) For $i = o(1)n - 1$ perform perfect line searches along the directions $d_i^{(o)}$, starting at $x_i^{(o)}$, such that $x_{i+1}^{(o)} = x_i^{(o)} + \alpha_i d_i^{(o)}$.

2) Set $d_n^{(o)} = x_n^{(o)} - x_o^{(o)}$, find the minimum z of $f(x)$ along $d_n^{(o)}$, starting at $x_n^{(o)}$: $z = x_n^{(o)} + \alpha_n d_n^{(o)}$.

3) Set $x_o^{(1)} = z$, discard $d_o^{(o)}$, set $d_o^{(1)} := d_1^{(o)}$,..., $d_{n-1}^{(1)} := d_n^{(o)}$, and start a new iteration.

A few comments may help to elucidate this procedure. Contrary to the Smith procedure all directions are used with equal weight. Moreover the complete space is covered already after n line searches, whereas Smith's method needs $\frac{1}{2}n(n+1)$ line searches. On the other hand, the Smith method finds the minimum of a quadratic function in one iteration, whereas Powell's first procedure requires n iterations as will be shown by induction. Note, however, that this is inherent in the definition of an iteration, still about twice as many line searches are necessary for Powell's method to locate the minimum (see below), as compared to Smith's method.

Assume that after k iterations the *new* directions $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$ are mutually conjugate. Start iteration (k+1) at $x_o^{(k)}$ and find the minimum u of the quadratic function $f(x)$ after successive line searches along $d_o^{(k)},\ldots,d_{n-k-1}^{(k)}$, starting at $x_o^{(k)}$. The subsequent line searches along $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$ will find, according to property vi) of paragraph 4.1., the minimum z in the k-dimensional space generated by u and the conjugate directions $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$. These directions have also been used to locate $x_o^{(k)}$, which is therefore a minimum of $f(x)$ with x in a "parallel subspace". The line joining z and $x_o^{(k)}$ is therefore, according to property vi), conjugate to $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$.

For the new iteration discard $d_o^{(k)}$, set $d_i^{(k+1)} := d_{i+1}^{(k)}$, $i = o(1)n - 2$, and $d_{n-1}^{(k+1)} = z - x_o^{(k)}$. Thus (k+1) conjugate directions have been generated after (k+1) iterations.

For $k = 2$ the arguments given above reduce to the simplified situation as described by property v) yielding a pair of conjugate directions which completes the proof by induction.

Example. Given $f(\xi,\eta) = \xi^2 + \eta^2 - \xi\eta - 2\xi - \eta + \frac{7}{3} \geq o$,

$$x^* = (\tfrac{5}{3}, \tfrac{4}{3})^T, \quad f(x^*) = o$$

$$x_o^{(o)} = (o,o)^T, \quad d_o^{(o)} = (1,o)^T, \quad d_1^{(o)} = (o,1)^T.$$

Powell's first procedure then produces the following sequence.

First iteration. Line search along $d_o^{(o)}$: $x_1^{(o)} = (1,o)^T$.

Line search along $d_1^{(o)}$: $x_2^{(o)} = (1,1)^T$.

Line search along $d_2^{(o)} = x_2^{(o)} - x_o^{(o)} = (1,1)^T$:

$h(\lambda) = f(x_2^{(o)} + \lambda d_2^{(o)}) = f(1 + \lambda, 1 + \lambda)$.

$h'(\lambda) = o$: $\lambda = \frac{1}{2}$, $x_o^{(1)} = (\frac{3}{2}, \frac{3}{2})^T$.

The new set of directions is given by

$$d_o^{(1)} = (o,1)^T , \quad d_1^{(1)} = (1,1)^T .$$

Second iteration.

Line search along $d_o^{(1)}$: $x_1^{(1)} = (\frac{3}{2}, \frac{5}{4})^T$.

Line search along $d_1^{(1)}$:

$h(\lambda) = f(x_1^{(1)} + \lambda d_1^{(1)}) = f(\frac{3}{2} + \lambda, \frac{5}{4} + \lambda)$.

$h'(\lambda) = o$: $\lambda = \frac{1}{8}$, $x_2^{(1)} = (\frac{13}{8}, \frac{11}{8})^T$.

Line search along $d_2^{(1)} = x_2^{(1)} - x_o^{(1)} = (\frac{1}{8}, -\frac{1}{8})^T$:

$h(\lambda) = f(x_2^{(1)} + \lambda d_2^{(1)}) = f(\frac{13}{8} + \frac{\lambda}{8}, \frac{11}{8} - \frac{\lambda}{8})$.

$h'(\lambda) = o$: $\lambda = \frac{1}{3}$, $x_o^{(3)} = (\frac{5}{3}, \frac{4}{3})^T$.

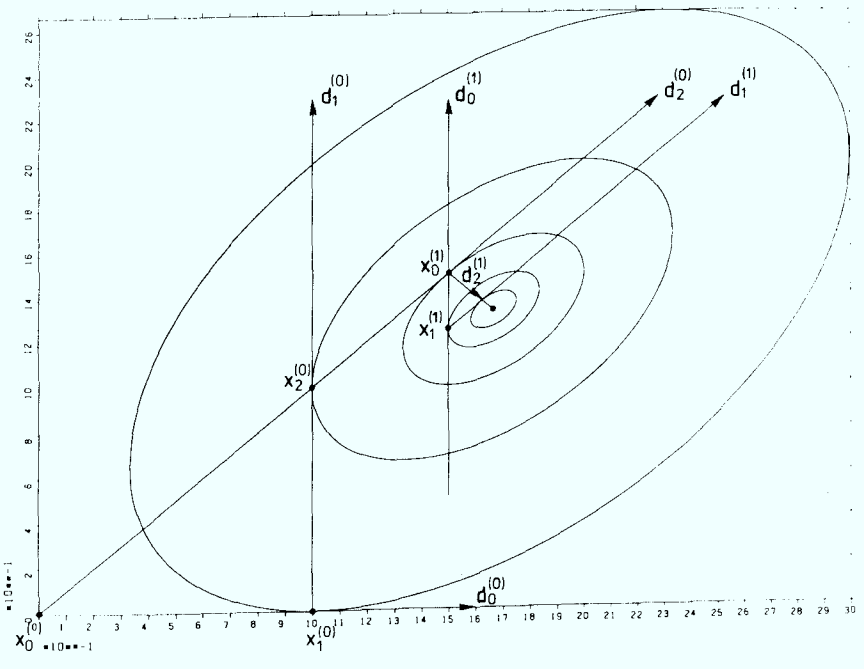


Fig. 4.2: Powell's first method for n=2.

(See Fig. 4.2 for this example). Each iteration requires (n+1) line searches to generate - and minimize along - one conjugate direction, the total number of line searches for minimizing a quadratic function is therefore given by $n(n+1)$, as against $\frac{1}{2}\, n(n+1)$ for Smith's method.

It has been observed (see Kowalik and Osborne |28|, p. 49, Fletcher |2o|, p. 77) that the first iteration should be replaced by just one minimization along $d_{n-1}^{(o)}$ yielding $x_o^{(1)}$. The second iteration should use the original directions $d_i^{(1)} \equiv d_i^{(o)}$, $i = o(1)n-1$. In this way the first conjugate direction, $d_{n-1}^{(1)} = d_{n-1}^{(o)}$, is obtained by only one line search thus reducing the total number of line searches to

$$1 + (n - 1)(n + 1) = n^2 \ .$$

This modified procedure is illustrated for $n=2$ in Fig. 4.3 (see the example below), and for $n=3$ in Fig. 4.4, where a schematic search path is indicated.

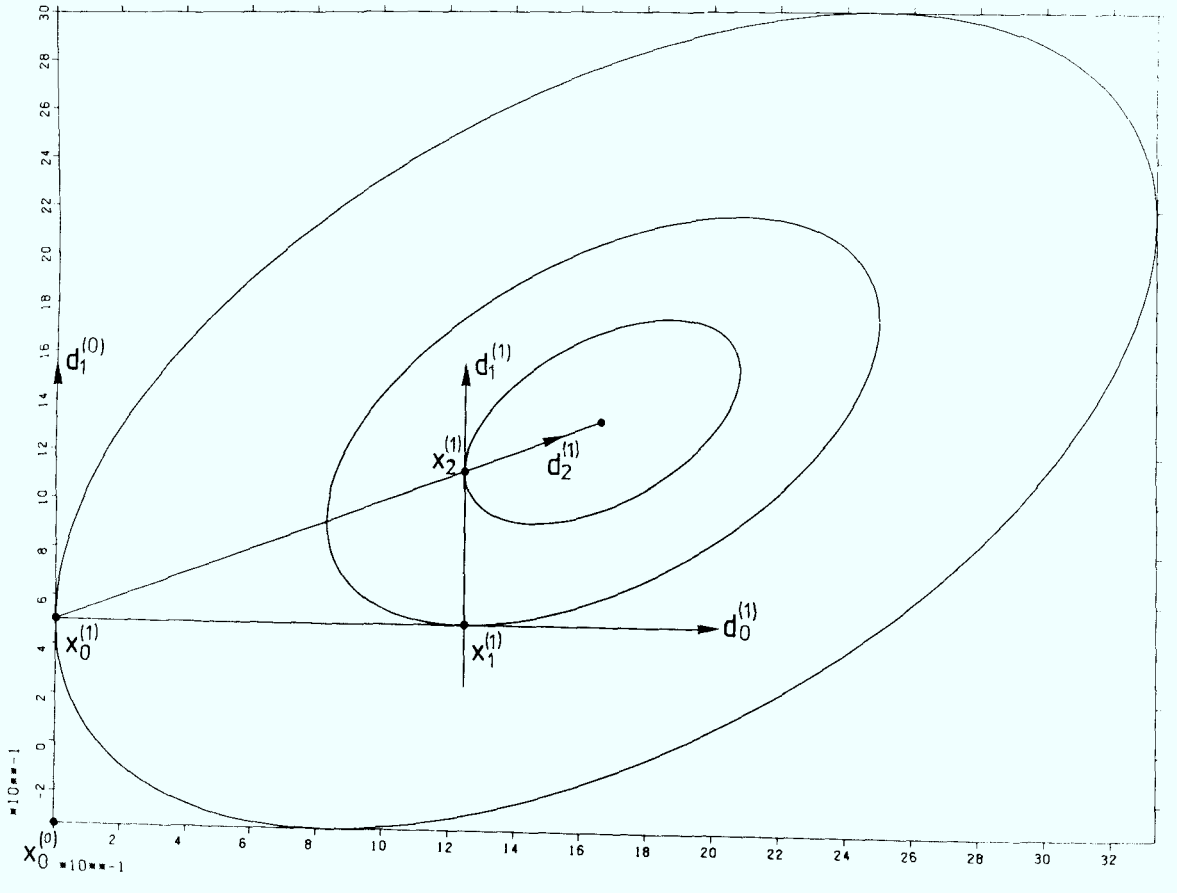


Fig. 4.3: Powell's modified first method for $n=2$.

Example (underlying Fig. 4.3)

Given $f(\xi,\eta)$, $x_o^{(o)}$, $d_o^{(o)}$, $d_1^{(o)}$ as in the preceding example.

Powell's modified first procedure then works as follows.

First iteration: Line search along $d_1^{(o)}$: $x_o^{(1)} = (o, \frac{1}{2})^T$.

Second iteration: Line search along $d_o^{(1)} \equiv d_o^{(o)}$: $x_1^{(1)} = (\frac{5}{4}, \frac{1}{2})^T$.

Line search along $d_1^{(1)} \equiv d_1^{(o)}$: $x_2^{(1)} = (\frac{5}{4}, \frac{9}{8})^T$.

Line search along $d_2^{(1)} = x_2^{(1)} - x_o^{(1)} = (\frac{5}{4}, \frac{5}{8})^T$:

$h(\lambda) = f(x_2^{(1)} + \lambda d_2^{(1)}) = f(\frac{5}{4} + \frac{5}{4}\lambda\ ,\ \frac{9}{8} + \frac{5}{8}\lambda)$.

$h'(\lambda) = o$: $\lambda = \frac{1}{3}$ $x_o^{(3)} = (\frac{5}{3}, \frac{4}{3})^T$.

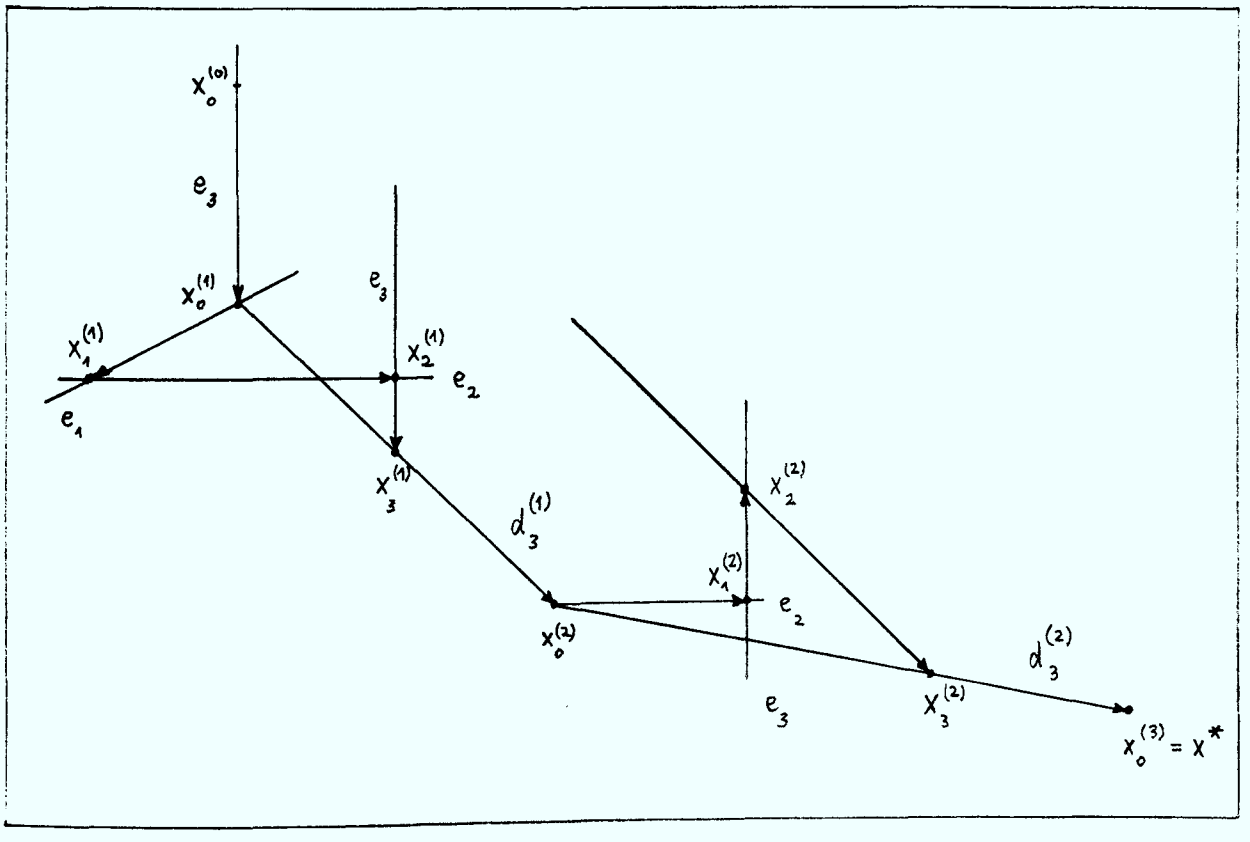


Fig. 4.4: Schematic path of Powell's modified first method for n=3 .

A final remark shall illustrate the connection between Smith's method and Powell's first procedure as pointed out by Fletcher [20]. When proving that Powell's method finds the minimum of a quadratic function in n iterations, the $(k+1)$-st iteration was split into two parts: First the minimization along $d_o^{(k)},\ldots,d_{n-k-1}^{(k)}$ moving $x_o^{(k)}$ to u, and then k linear searches along conjugate directions $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$, moving u to $z := x_n^{(k)}$. In Smith's terminology the first part corresponds to his arbitrary step β_k along d_k which displaces x_k to $u = x_k + \beta_k d_k$, and the second part to the minimization along $p_o,\ldots,p_{k-1}$, moving u to z. The analogy between both methods follows, if we set

$$p_i := d_{n-k+i}^{(k)}, \quad i = o(1)k - 1$$

(Note that e.g. $p_o := d_{n-k}^{(k)} = d_{n-k+1}^{(k-1)} = \ldots = d_{n-1}^{(1)}$.)

At the end of the $(k+1)$-st iteration both methods discard the first search direction, $d_o^{(k)}$ and d_k respectively, and add the new conjugate direction $p_k := d_n^{(k)} = z - x_o^{(k)}$. The first iteration of Smith's method is identical with Powell's *modified* first procedure, since then $p_o := d_{n-1}^{(1)} := d_{n-1}^{(o)}$.

We therefore conclude that Powell's method replaces Smith's arbitrary step by $(n-k)$ line searches and thus guarantees that the complete space is covered after each iteration, i.e. after n line searches.

A difficulty arising in Powell's first procedure will be discussed and removed when Zangwill's modification is introduced (see paragraph 4.4.).

4.3.2. Powell's second procedure

Powell's first procedure turned out to be rather inefficient when applied to larger problems. Powell observed that after some iterations the new directions tended to become nearly linearly dependent. From this weakness his second procedure originated which is still used as a standard algorithm in unconstrained minimization. The original presentation [39] is very concise, so we prefer to give a detailed description which follows mainly the Kowalik-Osborne line [28].

The algorithm first performs line searches along $d_o,\ldots,d_{n-1}$ moving x_o to x_n which therefore coincides with the first procedure. Then criteria are tested, whether or not to replace a direction. For this purpose the function value on the line of total progress, $d_n = x_n - x_o$, is calculated at $x_n + (x_n - x_o) = (2x_n - x_o)$. Set

$$f_1 := f(x_o)\ ,\quad f_2 := f(x_n)\ ,\quad f_3 := f(2x_n - x_o)\ ,$$

and proceed according to the following strategy explained later.

1) $f_3 \geq f_1$. Use old directions for the next iteration and x_n as new starting point.

2) $f_3 < f_1$. Determine

$$\Delta = \max_{1\leq i\leq n} \{f(x_{i-1}) - f(x_i)\} = f(x_{m-1}) - f(x_m)\ ,\ m \in \mathbb{N}[1,n]\ .$$

Distinguish the two cases:

a) $(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta)^2 \geq \frac{1}{2}\Delta(f_1 - f_3)^2$. Proceed as under 1), except that $(2x_n - x_o)$ should be used as a new starting point, if $f_3 < f_2$.

b) Calculate the minimum along $d_n = (x_n - x_o)$, which defines the new starting point for the next iteration, discard direction d_{m-1} and use the set of directions

$$d_o,\ldots,d_{m-2}\ ,\ d_m,\ldots,d_n\ ,\quad \text{if}\quad 2 \leq m \leq n\ ,$$

$$d_1,\ldots,d_{n-1}\ ,\ d_n\ ,\qquad\qquad \text{if}\quad m = 1\ .$$

As a consequence of this strategy it may happen in later iterations that a direction d_{m-1} is discarded which already belonged to the set of conjugate directions generated before. Therefore this algorithm may require more than n iterations to find the minimum of a quadratic function. In fact an upper bound for the number of iterations cannot be given meaning that Powell's second procedure does not possess the property of quadratic termination.

A derivation of Powell's criteria will be given now. Property vii) revealed that the determinant of the matrix $(d_o,\ldots,d_{n-1})$ with appropriately scaled d_i assumes its maximum value, if all d_i are conjugate. Powell's second procedure tries to keep this determinant as large as possible. The following considerations are valid for quadratic functions only, the resulting formulae will be applied to non-quadratic functions as well.

If x_n denotes the point after n perfect line searches, we can always assume that

$$x_n = x_o + \sum_{i=o}^{n-1} \alpha_i d_i \ , \qquad \text{with} \quad d_i^T G\, d_i = 1 \ , \tag{4.11}$$

$$d_n = x_n - x_o = \mu d \ , \qquad \text{with} \quad d^T G\, d = 1 \ . \tag{4.12}$$

Let D be the determinant of the matrix $(d_o,\ldots,d_{n-1})$. If we replace the direction d_{m-1} , that is the direction along which the maximum decrease in function value has occurred, by (assume for simplicity that $m > 1$)

$$d := \frac{1}{\mu} d_n = \frac{1}{\mu} \alpha_{m-1} d_{m-1} + \frac{1}{\mu} (\alpha_o d_o + \ldots + \alpha_{m-2} d_{m-2} + \alpha_m d_m + \ldots + \alpha_{n-1} d_{n-1}) \ ,$$

the new value of the determinant is $\frac{1}{\mu} \alpha_{m-1} D$. This follows because all terms proportional to d_i , $i = o(1)n-1$, $i \neq m-1$, in the expression for d do not contribute to the determinant. Consequently a replacement should only be made if

$$|\alpha_{m-1}| > \mu \ . \tag{4.13}$$

Note that the initial set of directions $d_o^{(o)},\ldots,d_{n-1}^{(o)}$ allows either sign of α_i, whereas only positive μ values will be of interest, since d_n is already a descent direction due to $f(x_o) > f(x_n)$.

For perfect line searches and quadratic functions $|\alpha_i|$ can be expressed explicitly by function values only. With $x_{i+1} = x_i + \alpha_i d_i$ and the scaling condition $d_i^T G\, d_i = 1$ we obtain

$$\alpha_i = d_i^T G(x_{i+1} - x_i) = \frac{1}{\alpha_i} (x_{i+1} - x_i)^T G(x_{i+1} - x_i) ,$$

or

$$\alpha_i^2 = (x_{i+1} - x_i)^T G(x_{i+1} - x_i) . \qquad (4.14)$$

Define

$$\Delta f_i := f(x_i) - f(x_{i+1}) = \frac{1}{2} (x_i^T Gx_i - x_{i+1}^T Gx_{i+1}) + b^T (x_i - x_{i+1}) .$$

The perfect line search along d_i allows to eliminate b. With $h(\lambda) = f(x_i + \lambda d_i)$:

$$h'(\lambda) = \lambda\, d_i^T Gd_i + d_i^T (Gx_i + b) .$$

The condition $h'(\alpha_i) = o$ together with the scaling property for the d_i leads to

$$\alpha_i = -d_i^T (Gx_i + b) ,$$

$$b^T (x_i - x_{i+1}) = -\alpha_i b^T d_i = \alpha_i (d_i^T Gx_i + \alpha_i)$$

$$= (x_{i+1} - x_i)^T Gx_i + \alpha_i^2 .$$

This inserted into Δf_i and using (4.14) finally gives

$$\Delta f_i = \frac{1}{2} \alpha_i^2 ,$$

$$|\alpha_i| = \sqrt{2\Delta f_i} = \sqrt{2(f(x_i) - f(x_{i+1}))} . \qquad (4.15)$$

Now μ, defined by (4.12), can be calculated. Let x_s denote the minimum along $d_n = \mu d$, and

$$f_s := f(x_s) , \quad f_1 := f(x_o) , \quad f_2 := f(x_n) , \quad f_3 := f(2x_n - x_o) .$$

We want to apply (4.15) to determine μ . This is possible because d is already properly scaled. Then x_s , the minimum along d , corresponds to x_{i+1} of (4.15). Two initial points x_i shall be considered. We find for

1) $x_i = x_o$: $\quad x_s = x_o + \sqrt{2(f_1 - f_s)}\, d$,

2) $x_i = x_n$: $\quad x_s = x_n \pm \sqrt{2(f_2 - f_s)}\, d$.

As $f(x_n) < f(x_o)$, the minimum of the quadratic function $f(x)$ along d , starting at x_o , must lie in the positive direction of d , but can be situated on either side of x_n .

a) $f_3 < f_1$. The positive sign in 2) must be chosen. Subtracting 1) from 2) yields

$$d_n = x_n - x_o = \{\sqrt{2(f_1 - f_s)} - \sqrt{2(f_2 - f_s)}\}\, d ,$$

and from (4.12)

$$\mu = \sqrt{2(f_1 - f_s)} - \sqrt{2(f_2 - f_s)} . \qquad (4.16)$$

b) $f_3 \geq f_1$. The negative sign in 2) must be chosen. This leads to

$$\mu = \sqrt{2(f_1 - f_s)} + \sqrt{2(f_2 - f_s)} . \qquad (4.17)$$

In this case:

$$\sqrt{f_1 - f_s} + \sqrt{f_2 - f_s} \equiv \sqrt{f_1 - f_2 + f_2 - f_s} + \sqrt{f_2 - f_s}$$

$$\geq \sqrt{f_1 - f_2} > \sqrt{f(x_{m-1}) - f(x_m)} = \frac{1}{\sqrt{2}} |\alpha_{m-1}|$$

using (4.15). The inequalities follow because $f_2 \geq f_s$, and

$$f_1 - f_2 \equiv f(x_o) - f(x_n) > \max_{1 \leq i \leq n} \{f(x_{i-1}) - f(x_i)\} .$$

Thus we have derived Powell's first criterion: If $f_3 \geq f_1$, the old directions should be retained, since then $\mu > |\alpha_{m-1}|$ which violates (4.13).

To derive the second criterion we need formula (2.21) which is valid for quadratic interpolation and can therefore be applied here, since only quadratic functions $f(x)$ are considered:

$$f_s = f_2 - \frac{1}{8} \frac{(f_1 - f_3)^2}{(f_1 - 2f_2 + f_3)} \quad . \tag{2.21}$$

Here $(f_1 - 2f_2 + f_3) > 0$, because $f(x)$ is assumed to be a positive definite quadratic form. Only the case $f_3 < f_1$ needs to be considered. Then μ follows from (4.16), and (4.13) is violated again if

$$|\alpha_{m-1}| \leq \sqrt{2(f_1 - f_s)} - \sqrt{2(f_2 - f_s)} \quad ,$$

or with $\Delta := f(x_{m-1}) - f(x_m)$:

$$\sqrt{\Delta} \leq \sqrt{f_1 - f_s} - \sqrt{f_2 - f_s} \quad .$$

This can be expressed as

$$f_1 - f_s \geq f_2 - f_s + \Delta + 2\sqrt{\Delta(f_2 - f_s)} \quad ,$$

$$(f_1 - f_2 - \Delta)^2 \geq 4\Delta(f_2 - f_s) \quad .$$

Inserting (2.21) finally gives Powell's second criterion: If

$$(f_1 - f_2 - \Delta)^2 \, (f_1 - 2f_2 + f_3) \geq \tfrac{1}{2}\Delta(f_1 - f_3)^2 \quad ,$$

and if $f_3 < f_1$, use again the old set of directions.

Powell's second procedure may use certain directions repeatedly. Therefore, the first time a line search is performed, the three function values for the quadratic approximation allow an estimate of the second derivative. With this estimate formula (2.22) can be used in further line searches thereby saving one function evaluation.

4.4. The method of Zangwill

Analyzing Powell's *first procedure*, Zangwill |55| observed that if the initial approximation x_o is such that the first line search along d_o does not yield a displacement, i.e. $\alpha_o = o$, then the directions for the following iteration are linearly dependent. This follows from

$$x_n = x_o + \sum_{i=1}^{n-1} \alpha_i \, d_i \,, \tag{4.18}$$

thus $d_n = (x_n - x_o)$ does not contain d_o, and the new set of directions $d_1,\ldots,d_n$ is linearly dependent.

With Powell's *second procedure* this cannot happen, since a direction d_i would not be replaced if $\alpha_i = o$, because then condition (4.13) is not satisfied. Zangwill's arguments concerning Powell's second procedure are wrong. Take e.g. his second counter-example. Let

$$f(\xi,\eta,\zeta) = (\xi-\eta+\zeta)^2 + (-\xi+\eta+\zeta)^2 + (\xi+\eta-\zeta)^2 \,,$$

let $x_o = (1/2,1,1/2)^T$ be the point reached after the first iteration and let e_1, e_2, e_3 be the search directions for the second iteration. Then $x_3 = (o,-2/3,-2/9)^T$ and

$$f_3 := f(2x_3 - x_o) \cong 3.o18 \,.$$

With $f_1 := f(x_o) = 2$ we have $f_3 > f_1$, and the coordinate directions are again chosen for the third iteration, starting at x_3. Thus Zangwill's second counter-example is meaningless.

Zangwill suggested a method which is essentially based on Powell's first procedure. Let $e_1,\ldots,e_n$ denote the coordinate directions, $d_o^{(k)},\ldots,d_{n-1}^{(k)}$ the search directions for iteration (k+1), along which $x_o^{(k)}$ is shifted to $x_n^{(k)}$ according to Powell's original method. The new direction, however, is no longer $x_n^{(k)} - x_o^{(k)}$, but

$$d_n^{(k)} = x_n^{(k)} - x_{n+1}^{(k-1)} \,, \quad k \geq 1 \,, \tag{4.19}$$

where $x_{n+1}^{(k-1)}$ is the minimum along $d_n^{(k-1)}$. The first iteration is the same as in Powell's modified procedure, i.e. $x_{n+1}^{(o)} = x_o^{(o)} + \alpha_{n-1}^{(o)} \, d_{n-1}^{(o)}$.

After $x_{n+1}^{(k)}$ has been determined, a search along the e_i is tried in turn, until the first non-zero shift occurs. This point defines $x_o^{(k+1)}$. Note that for Powell's first procedure we have $x_o^{(k+1)} = x_{n+1}^{(k)}$, whereas now $x_o^{(k+1)} \neq x_{n+1}^{(k)}$. The search along the e_i starts with that direction which follows the one, along which the preceding iteration found the non-zero shift. If after n consecutive searches along the e_i no shift is found, the minimum has been located at $x_{n+1}^{(k)}$.

In order to get some insight into Zangwill's method, let us consider again the dependence of d_n on d_o, the direction to be deleted. Using (4.19) we have

$$d_n^{(k)} = \sum_{i=o}^{n-1} \alpha_i^{(k)} d_i^{(k)} + x_o^{(k)} - x_{n+1}^{(k-1)} ,$$

where $x_o^{(k)} - x_{n+1}^{(k-1)}$ is by construction non-vanishing. We distinguish two cases: The shift is either parallel to $d_o^{(k)}$ or not.

a) $x_o^{(k)} - x_{n+1}^{(k-1)} = \beta_o d_o^{(k)}$, where $d_o^{(k)}$ must be parallel to one of the e_i and $\beta_o \neq o$ is obtained from a perfect line search along $d_o^{(k)}$. Therefore $\alpha_o^{(k)} = o$ in the succeeding search along $d_o^{(k)}$, and the vector

$$d_n^{(k)} = \sum_{i=1}^{n-1} \alpha_i^{(k)} d_i^{(k)} + \beta_o d_o^{(k)}$$

contains a non-vanishing contribution of $d_o^{(k)}$, which guarantees the linear independence of the new set

$$\{d_o^{(k+1)},\ldots,d_{n-1}^{(k+1)}\} := \{d_1^{(k)},\ldots,d_n^{(k)}\} .$$

b) $x_o^{(k)} - x_{n+1}^{(k-1)}$ is not parallel to $d_o^{(k)}$. Then it may happen that $\alpha_o^{(k)} + c_o^{(k)} = o$, where $c_o^{(k)}$ is the component of $x_o^{(k)} - x_{n+1}^{(k-1)}$ along $d_o^{(k)}$ when decomposing the shift with respect to $d_i^{(k)}$, $i = o(1)n-1$. Whenever $\alpha_o^{(k)} + c_o^{(k)} = o$, the directions are linearly dependent.

The following example illustrates this last remark.

Example Given $f(\xi,\eta,\zeta) = \xi^2 + \eta^2 + \zeta^2 + \xi\eta - 2\xi - 3\eta + \frac{7}{3} \geq o$,

$$f(\tfrac{1}{3}, \tfrac{4}{3}, o) = o , \quad x_o^{(o)} = (o,o,1)^T$$

$$d_o^{(1)} = (-1,1,o)^T \ , \ d_1^{(1)} = (o,1,o)^T \ , \ d_2^{(1)} = (o,o,1)^T$$

Zangwill's method then proceeds as follows.

First iteration. Line search along $d_2^{(o)} \equiv d_2^{(1)}$: $x_4^{(o)} = (o,o,o)^T$

Second iteration. Line search along e_1 : $x_o^{(1)} = (1,o,o)^T$

Line search along $d_o^{(1)}$:

$h(\lambda) = f(x_o^{(1)} + \lambda d_o^{(1)}) = f(1 - \lambda, \lambda, o)$.

$h'(\lambda) = o$: $\lambda = 1$, $x_1^{(1)} = (o,1,o)^T$

Line search along $d_1^{(1)}$: $x_2^{(1)} = (o,\frac{3}{2},o)^T$

Line search along $d_2^{(1)}$: $x_3^{(1)} = x_2^{(1)}$

Line search along $d_3^{(1)} = x_3^{(1)} - x_4^{(o)}$: $x_4^{(1)} = x_2^{(1)}$

The new set of directions is *linearly dependent*:

$$d_o^{(2)} = (o,1,o)^T \ , \ d_1^{(2)} = (o,o,1)^T \ , \ d_2^{(2)} = (o,\tfrac{3}{2},o)^T \ .$$

Third iteration. Line searches along e_2, e_3 : $x_4^{(1)} = x_2^{(1)}$.

Line search along e_1 : $x_o^{(2)} = (\frac{1}{4},\frac{3}{2},o)^T$

Line search along $d_o^{(2)}$: $x_1^{(2)} = (\frac{1}{4},\frac{11}{8},o)^T$

Line searches along $d_1^{(2)}$, $d_2^{(2)}$: $x_3^{(2)} = x_1^{(2)}$

Line search along $d_3^{(2)} = x_3^{(2)} - x_4^{(1)} = (\frac{1}{4},-\frac{1}{8},o)^T$:

$h(\lambda) = f(x_3^{(2)} + \lambda d_3^{(2)}) = f(\frac{1}{4}(1+\lambda), \frac{1}{8}(11-\lambda), o)$.

$h'(\lambda) = o$: $\lambda = \frac{1}{3}$, $x_4^{(2)} = (\frac{1}{3},\frac{4}{3},o)^T$.

Thus the exact minimum is found after $n = 3$ iterations. The (conjugate) directions after the third iteration are (up to a scaling factor):

$$d_o^{(3)} = (o,o,1)^T \ , \ d_1^{(3)} = (o,1,o)^T \ , \ d_2^{(3)} = (2,-1,o)^T \ .$$

Initially a special set of *linearly independent directions* $d_i^{(1)}$, $i = o(1)n - 1$, was chosen, which then became linearly dependent after the second iteration. The choice of initial directions is, however, completely insignificant in what concerns the process of successive generation of conjugate directions. This will become clear hereafter. Important is only the injection of coordinate directions to avoid premature termination. Fig. 4.5 illustrates the path of minimization as projected onto the plane $\zeta = o$ for the example given before.

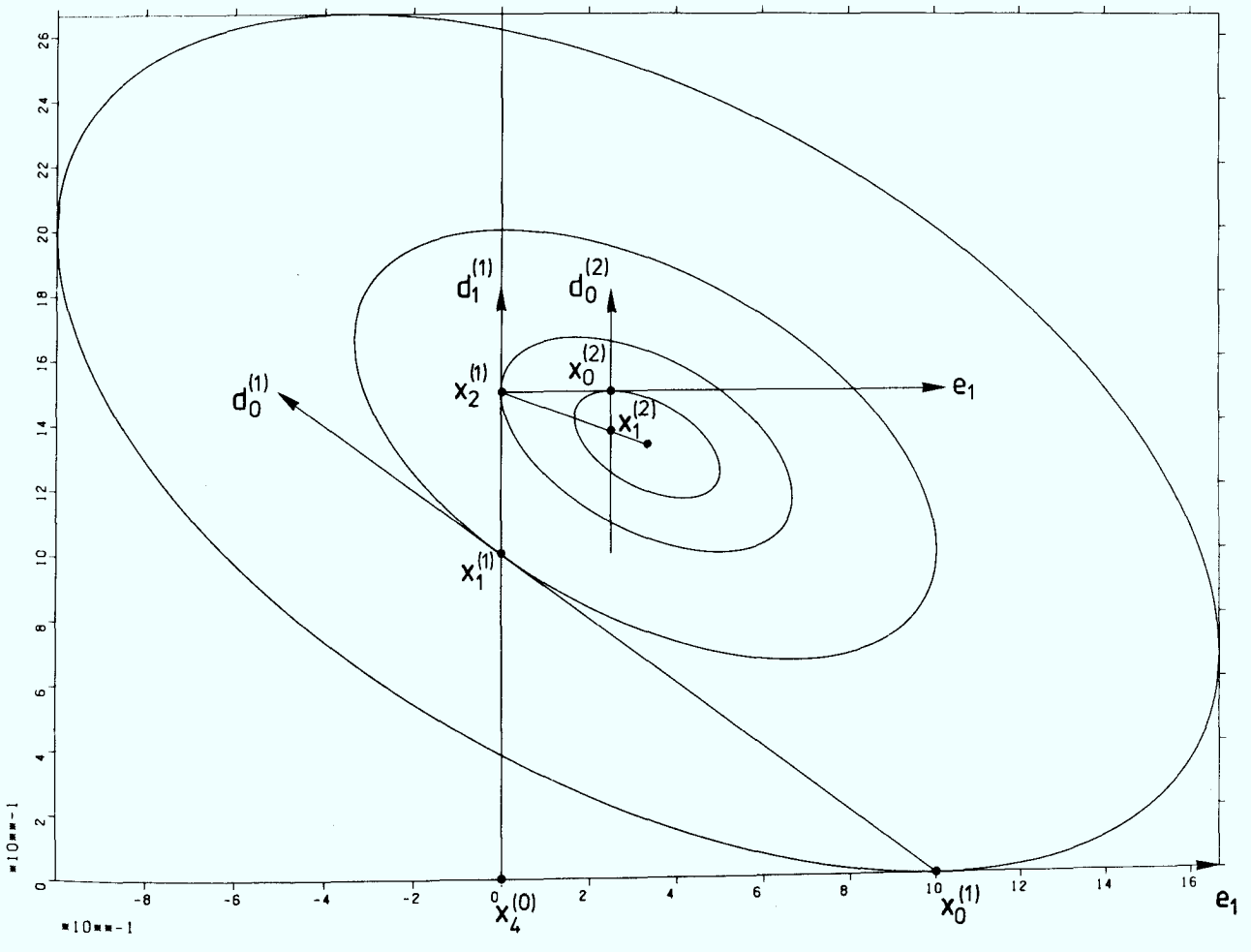


Fig. 4.5: Zangwill's method after the first iteration.

As can be conjectured from this example, Zangwill's method will possess the property of quadratic termination. The easiest way to show this property is again the interpretation of Zangwill's method as a special case of Smith's method.

To this end, iteration (k+1) is divided into two parts:

1) A non-zero shift from $x_{n+1}^{(k-1)}$ to $x_o^{(k)}$, and (n-k) consecutive line searches along $d_o^{(k)},\ldots,d_{n-k-1}^{(k)}$ moving $x_o^{(k)}$ to u.
2) k line searches along $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$ moving u to $x_n^{(k)}$, and subsequent minimization along $d_n^{(k)} = x_n^{(k)} - x_{n+1}^{(k-1)}$ yielding $x_{n+1}^{(k)}$.

By induction assume that $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$ are mutually conjugate (and of course non-zero). Then $x_{n+1}^{(k-1)}$, which was found in the previous iteration after minimization along these directions, is the minimum of the k-dimensional space S_1 spanned by these directions and containing a certain starting point v, say. This follows again from property vi) and its remark (see paragraph 4.1.). Similarly the point $x_n^{(k)}$ is the minimum in the space S_2 spanned by the same set of directions, but starting at u. Now u cannot be a point in S_1, since the initial starting point $x_o^{(k)}$ is already not in S_1, which follows because $x_{n+1}^{(k-1)}$ is the minimum in S_1, and $f(x_o^{(k)}) < f(x_{n+1}^{(k-1)})$ by construction. Although u is found by line searches as well, we cannot exclude that some or all $\alpha_i^{(k)}$, $i = o(1)n-k-1$, are zero. Therefore only $f(u) \leq f(x_o^{(k)})$ holds which, however, is sufficient for u not in S_1. From property vi), the line joining the two minima, $d_n^{(k)} = x_n^{(k)} - x_{n+1}^{(k-1)}$, is conjugate to $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$. Note that $d_n^{(k)}$ cannot vanish, as $x_{n+1}^{(k-1)} \varepsilon S_1$, $x_n^{(k)} \varepsilon S_2$, and both spaces have no common point.

For $k=1$ just one conjugate direction, $d_{n-1}^{(1)} \equiv d_{n-1}^{(o)}$, is given by construction. This completes the proof that Zangwill's method finds the minimum of a quadratic function in at most n iterations.

A few remarks should be added. First of all Zangwill's method avoids the main drawbacks of Powell's first and second procedure, which were : premature termination for quadratic functions if linear dependence of search directions occurs (see section 4.3.1.), and absence of the property of quadratic termination (see section 4.3.2.) On the other hand, the injection of line searches along coordinate

directions can become expensive: This number of line searches is at least one, at most n, per iteration. Quadratic termination is therefore achieved after n_ℓ line searches, where $1 + (n-1)(n+2) \leq n_\ell \leq 1 + (n-1)(2n+1)$. In the example given above: $n_\ell = 13$, as against 9 line searches for Powell's modified first procedure for $n = 3$. This is the expense at which Zangwill's method is safer. One should keep in mind, however, that for non-quadratic problems also the additional line searches always contribute to the reduction in function value.

4.5. On quadratic termination of Powell's modified first procedure

It is very instructive to consider the problem of quadratic termination of Powell's modified first procedure in view of the proof of quadratic termination for Zangwill's method. Again the Smith approach will be adopted. Then iteration (k+1) splits into the two parts:

1) $u^{(k)} = x_o^{(k)} + \sum_{i=o}^{n-k-1} \alpha_i^{(k)} d_i^{(k)}$,

2) $x_n^{(k)} = u^{(k)} + \sum_{i=n-k}^{n-1} \alpha_i^{(k)} d_i^{(k)}$, and minimization along

$d_n^{(k)} = x_n^{(k)} - x_o^{(k)}$ leads to $x_o^{(k+1)}$. The new directions are

$d_i^{(k+1)} = d_{i+1}^{(k)}$, $i = o(1)n-1$.

As before, $S_1^{(k)}$ and $S_2^{(k)}$ are two spaces spanned by $d_{n-k}^{(k)}, \ldots, d_{n-1}^{(k)}$ and containing $x_o^{(k)}$ (which is the minimum in $S_1^{(k)}$) and $u^{(k)}$, respectively. The condition that the two spaces are different amounts to requiring:

$$\begin{array}{lll} \text{for iteration } 2: & x_o^{(1)} + \alpha_o^{(1)} d_o^{(1)} + \ldots + \alpha_{n-2}^{(1)} d_{n-2}^{(1)} & \text{not in } S_1^{(1)} \\ \vdots & \vdots & \vdots \\ \text{for iteration } n: & x_o^{(n-1)} + \alpha_o^{(n-1)} d_o^{(n-1)} & \text{not in } S_1^{(n-1)}. \end{array} \qquad (4.2o)$$

Necessary and sufficient conditions are: For each k, $k = 1(1)n-1$,

at least one $\alpha_i^{(k)}$, i = o(1)n-k-1 , is not equal zero. Then the corresponding $u^{(k)}$ is not in $S_1^{(k)}$, because the $\alpha_i^{(k)}$ are obtained by perfect line searches and thus reduce the function value when moving from $x_o^{(k)}$, the minimum in $S_1^{(k)}$.

One possible way to realize these conditions is: $\alpha_o^{(k)} \neq o$, k = 1(1)n-1 (see (4.2o)). *In addition* this choice guarantees linearly independent directions $d_o^{(k+1)},\ldots,d_{n-1}^{(k+1)}$, supposing that the directions $d_o^{(k)},\ldots,d_{n-1}^{(k)}$ form a set of linearly independent vectors: The new direction $d_{n-1}^{(k+1)} := d_n^{(k)}$ contains a non-vanishing component of $d_o^{(k)}$, that is the direction to be deleted for the new iteration. Since the initial set of directions is linearly independent, this holds for all iterations.

If on the other hand $\alpha_o^{(k)} = o$ for some k , then the subsequent iteration - and all following ones - will limit the search to the (n-1)-dimensional space not containing $d_o^{(k)}$. The following example illustrates this behaviour.

Example Given the same function as in the previous paragraph. Further:

$$x_o^{(o)} = (1,o,1)^T ,$$

$$d_o^{(o)} = (1,o,o)^T , \; d_1^{(o)} = (o,1,o)^T , \; d_2^{(o)} = (o,o,1)^T .$$

Powell's modified first procedure then works as follows (note that $d_i^{(1)} \equiv d_i^{(o)}$, i = o,1,2).

First iteration. Line search along $d_2^{(o)}$: $x_o^{(1)} = (1,o,o)$

Second iteration. Line search along $d_o^{(1)}$: $x_1^{(1)} = x_o^{(1)}$.

Line search along $d_1^{(1)}$: $x_2^{(1)} = (1,1,o$

Line search along $d_2^{(1)}$: $x_3^{(1)} = x_2^{(1)}$.

Line search along $d_3^{(1)} = x_3^{(1)} - x_o^{(1)}$: $x_o^{(2)} = x_3^{(1)}$.

The new set of directions is linearly dependent:

$$d_o^{(2)} = (o,1,o)^T \ , \ d_1^{(2)} = (o,o,1)^T \ , \ d_2^{(2)} = (o,1,o)^T \ . \quad (4.21)$$

Third iteration. The algorithm stops at $x_o^{(3)} = (1,1,o)^T$.

If the second iteration of this example is divided according to the Smith scheme given above, we find

$$u^{(1)} := x_o^{(1)} + \alpha_o^{(1)} \ d_o^{(1)} + \alpha_1^{(1)} \ d_1^{(1)} = x_o^{(1)} + d_1^{(1)} \ .$$

A new *conjugate direction*, $d_2^{(2)} := d_3^{(1)}$, is generated after this iteration, because $\alpha_1^{(1)} \neq o$ and the necessary and sufficient condition derived above is therefore satisfied. But $\alpha_o^{(1)} = o$, and the set of directions (4.21) , which contains two conjugate ones, becomes linearly dependent.

If we try to separate the third iteration as well, we obtain

$$u^{(2)} := x_o^{(2)} + \alpha_o^{(2)} \ d_o^{(2)} = x_o^{(2)} \ .$$

The necessary and sufficient condition for the generation of a further conjugate direction is violated and the algorithm terminates prematurely.

With Zangwill's method this will never happen. When arriving at $x_{n+1}^{(k)}$, the complete space is still available, no matter whether the $d_i^{(k)}$ are linearly dependent or not.

With this paragraph we tried to elucidate, that the conditions $\alpha_o^{(k)} \neq o$, $k = 1(1)n-1$, are not necessary to generate new conjugate directions (except for $k = n-1$), but are necessary and sufficient to maintain linear independence. Recently Powell [41] stated that quadratic termination can be shown provided that $\alpha_o^{(k)} \neq o$, without giving details for this particular assumption. This initiated the investigation into this problem in more detail.

4.6. Some other algorithms without derivatives

Although Powell's second procedure, known as the *Powell 64 method*, dates back to 1964, only a few improvements of Powell's basic approach and some new derivative-free methods have been designed till then. We give a short description of the methods of Brent |8| Sutti |53| and Brodlie | 9 | and mention the method of Chazan and Miranker |32|, which is a parallel algorithm. A computer code is available only for Brent's method: an ALGOL W listing in his book.

4.6.1. The method of Brent

Brent's method |8| is mainly based on Powell's *modified first procedure*, but uses also features of Powell's *second procedure*, with new ideas as well. It belongs to the class of *restart* or *reset algorithms*: After a fixed number of iterations, usually n, the algorithm restarts cyclicly with an improved starting vector x_o, and therefore deletes most of the information gained so far.

Every restart of Brent's method retains some information which for quadratic functions would be most desirable: It starts each new cycle of n iterations along the *principal axes* of the quadratic approximation to the objective function, which is defined through the Hessian matrix $G_{\ell j} := \partial^2 f / \partial \xi_\ell \, \partial \xi_j$ at the actual point (we know from property ii) of paragraph 4.1., that eigenvectors form a particular set of conjugate directions). In addition, the *orthogonality* of the eigenvectors offers optimal linear independence conditions for the new cycle.

In order to avoid the calculation of the second derivatives, Brent suggested an approximation for G which originates from the quadratic case. Moreover, it turns out that it is much easier to obtain G^{-1} than G, which, however, is sufficient since not G itself is required, but only its eigenvectors which are equal to those of G^{-1}.

Let us consider the quadratic case. Then we know that Powell's first procedure provides n conjugate directions after n iterations. With $U := (p_o, \ldots, p_{n-1})$ the inverse of the matrix

G underlying the quadratic function can be expressed using property iv) of paragraph 4.1.:

$$G^{-1} = \sum_{i=o}^{n-1} \frac{p_i p_i^T}{p_i^T G p_i} \equiv U \Gamma^{-1} U^T , \qquad (4.22)$$

where $\Gamma = \mathrm{diag}(\gamma_i)$, $\gamma_i = p_i^T G p_i$.

The diagonal elements γ_i can be determined directly by taking second differences of $h_i(\lambda) = f(x_i + \lambda p_i)$, which in the quadratic case reduces to

$$h_i(\lambda) = \frac{1}{2} \lambda^2 \gamma_i + \lambda p_i^T (G x_i + b) + f(x_i) .$$

At least three function values are known anyway from the line search along p_i . The additional computational effort is therefore relatively small to calculate G^{-1} as against the *direct computation* of G requiring $\frac{1}{2} n(n+1)$ second differences; but it should be realized that for non-quadratic functions this is only an approximation for G^{-1} , since the directions p_i are no longer conjugate.

Once G^{-1} is known, its eigenvectors are determined by a most efficient procedure. Let Q be the orthogonal matrix which diagonalizes G :

$$G = Q \Lambda Q^T , \quad \Lambda = \mathrm{diag}(\lambda_i) .$$

Then with (4.22):

$$G^{-1} = Q \Lambda^{-1} Q^T = U \Gamma^{-1} U^T .$$

Let $V := U \Gamma^{-1/2}$. Obviously

$$G^{-1} = VV^T = Q \Lambda^{-1/2} R^T R \Lambda^{-1/2} Q^T ,$$

where an arbitrary orthogonal matrix R has been inserted - usually the columns of R are the eigenvectors of V^TV . Then we can write:

$$V = Q \Lambda^{-1/2} R^T ,$$

which is known as the *singular value decomposition* of V .

Ideally Brent's method would work as follows. Perform n iterations according to Powell's modified first procedure. Then calculate $V = U \Gamma^{-1/2}$ and apply the singular value decomposition to V (choose e.g. subroutine LSVALR from IMSL). Note that V is much better conditioned than $G^{-1} = VV^T$. The new set of directions is given by the columns of Q.

In order to avoid the difficulties encountered with Powell's first procedure, Brent also attempts to maximize the determinant

$$|\det(d_o^{(k)},\ldots,d_{n-1}^{(k)})| ,$$

where the $d_i^{(k)}$ are appropriately scaled. But directions are discarded only from the set of *non-conjugate directions* $d_i^{(k)}$, $i = o(1)n-k-1$, and the new direction $d_n^{(k)}$ is added in any case. Thus the property of quadratic termination, violated in Powell's second procedure, is preserved.

The problem of linear dependence (see paragraph 4.5.) still remains and Brent suggests to try random steps along the $d_i^{(k)}$, $i = o(1)n-1$, at the beginning of the $(k+1)$-st iteration until a decrease in function value is obtained. This is similar to Zangwill's search along the coordinate directions (see paragraph 4.4.). But whereas Zangwill's method allows temporarily linearly dependent directions (see the example of paragraph 4.4.), this would prevent the application of the discarding criterion of Brent's method and, what is more serious, would restrict the random search of the next iteration to a subspace, from which the method will not recover. The random steps of Brent's method therefore must be performed such that the component along the direction to be discarded does not vanish, where this component consists of a random part and a line search part. Then linear independence and thereby quadratic termination will be ensured.

Finally we quote a remark by Sutti, see |52|, p. 295, where she points out correctly that the determination of Q and Λ requires theoretically infinitely many steps.

> "Therefore we conclude that Brent's method minimizes a quadratic in a finite number n^2 of linear searches, but not theoretically in a finite number of operations."

Applying Brent's method to a quadratic function finds its minimum after at most n iterations, *before* the diagonalization is applied for the first time.

4.6.2. The method of Sutti

This method (|53|, pp. 277) is best described if again the Smith approach is adopted. For simplicity we limit the considerations to quadratic functions. One cycle of n iterations then proceeds as follows. Given n linearly independent directions $d_o^{(1)}, \ldots, d_{n-1}^{(1)}$, and $x_o^{(o)}$.

1) First iteration. Minimize along $d_{n-1}^{(1)}$ yielding $x_o^{(1)}$ (as in Powell's modified first procedure).
2) For iteration $k+1$, $k = 1(1)n-1$, try the *non-conjugate* directions $d_o^{(k)}, \ldots, d_{n-k-1}^{(k)}$ in turn by stepping with arbitrary step-lengths $\beta_j^{(k)} > o$, until a decrease in function value occurs for the first time:

$$f(x_o^{(k)} \pm \beta_j^{(k)} d_j^{(k)}) < f(x_o^{(k)}), \qquad j \in \mathbb{I}[o,n-k-1].$$

If this condition cannot be satisfied, try line searches along the same set of directions until for the first time

$$f(x_o^{(k)} + \alpha_j^{(k)} d_j^{(k)}) < f(x_o^{(k)}), \qquad j \in \mathbb{I}[o,n-k-1].$$

If this also fails, the algorithm stops. Otherwise set

$$u = x_o^{(k)} \pm \beta_j^{(k)} d_j^{(k)}, \quad \text{or} \quad u = x_o^{(k)} + \alpha_j^{(k)} d_j^{(k)},$$

respectively.
3) Perform line searches along the *conjugate* directions $d_{n-k}^{(k)}, \ldots, d_{n-1}^{(k)}$, moving u to $x_n^{(k)}$. Set $d_n^{(k)} = x_n^{(k)} - x_o^{(k)}$, and find the minimum $x_o^{(k+1)}$ along $d_n^{(k)}$.
4) The new directions are

$$\{d_o^{(k+1)}, \ldots, d_{n-1}^{(k+1)}\} = \{d_o^{(k)}, \ldots, d_{j-1}^{(k)}, d_{j+1}^{(k)}, \ldots, d_n^{(k)}\},$$

i.e. the direction $d_j^{(k)}$, along which the function decrease of step 2) occurred, has been deleted.

Remarks

i) If for all k in step 2) the arbitrary step is always successful, a cycle of n iterations requires only $\frac{1}{2}\, n(n+1)$ line searches.

ii) If, moreover, for all k in step 2) the *first arbitrary step* is always successful, Sutti's method is identical with Smith's method. Therefore, the weak points of the latter method (see paragraph 4.2.) will be shared to some extent by Sutti's method, i.e. each iteration does in general not cover the complete space.

iii) The other extreme case would be: For all k in step 2) the *last possible line search* will be successful. Then steps 1) to 3) are formally identical with Powell's modified first procedure, however in step 4) the direction $d_{n-k-1}^{(k)}$ is rejected thus avoiding linear dependence.

iv) Step 2) ensures that a new conjugate direction will always be generated (see paragraph 4.5.). Whenever this step fails to produce a function decrease, a search along $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$, i.e. step 3), would be meaningless since $x_o^{(k)}$ is already the minimum of the subspace spanned by $d_{n-k}^{(k)},\ldots,d_{n-1}^{(k)}$ and containing the point u of the previous iteration; therefore, if $d_o^{(k)},\ldots,d_{n-1}^{(k)}$ are linearly independent the algorithm must have found the minimum $x^* = x_o^{(k)}$.

v) As with Brent's method the search directions have to remain linearly independent so that the search in step 2) allows to cover the complete subspace complementary to the one mentioned in remark iv). If the directions $d_o^{(k)},\ldots,d_{n-1}^{(k)}$ are linearly independent, deleting $d_j^{(k)}$ and adding

$$d_n^{(k)} = \gamma_j^{(k)}\, d_j^{(k)} + \sum_{i=n-k}^{n-1} \alpha_i^{(k)}\, d_i^{(k)} ,$$

where $\gamma_j^{(k)}$ is given by either $\pm\, \beta_j^{(k)} \neq o$ or $\alpha_j^{(k)} \neq o$, will maintain linear independence of $d_o^{(k)},\ldots,d_{j-1}^{(k)}$, $d_{j+1}^{(k)},\ldots,d_n^{(k)}$.

Remarks iv) and v) ensure that Sutti's method possesses the property of quadratic termination.

4.6.3. The method of Brodlie

All methods discussed so far in this chapter originated from the method of Smith. Brodlie's algorithm |9| is completely different. It uses the fact that for quadratic functions the eigenvectors of the Hessian are conjugate directions (property ii) of paragraph 4.1.). The Hessian is symmetric, therefore a Jacobi type diagonalization can be applied.

For quadratic functions Brodlie proves that his algorithm is identical with a special version of Jacobi's method to determine the eigenvalues and eigenvectors of a symmetric matrix. As the latter method requires infinitely many iterations it is clear that Brodlie's method does not exhibit the property of quadratic termination. Sutti |52| points out that the Jacobi method becomes poor for high-dimensional matrices, so this property is reflected to some extent in the numerical results given by Brodlie. Still the algorithm works surprisingly well for small problems.

We present a short outline of this interesting strategy. Initially we assume that a set of n orthonormal directions $d_o^{(o)},\ldots,d_{n-1}^{(o)}$ - usually the coordinate directions - and a starting point x_o with $f(x_o)$ are given. Iteration $k+1$, $k = o,1,\ldots$, then works as follows:

1) Select a pair of directions $d_i^{(k)}$, $d_j^{(k)}$, $i \neq j$, and carry out a rough line search along these two directions in the following way. Calculate $f(x)$ at two points on each direction, thus together with $f(x_k)$ a quadratic interpolation is possible for both directions passing through x_k.

2) Consider the function $h(\lambda,\mu)$ defined over the plane spanned by $d_i^{(k)}$, $d_j^{(k)}$ and containing x_k:

$$h(\lambda,\mu) = f(x_k + \lambda d_i^{(k)} + \mu d_j^{(k)})$$

$$= \frac{1}{2}\left[a\lambda^2 + 2b\lambda\mu + c\mu^2\right] + d\lambda + e\mu + f(x_k) ,$$

where it has been used that $f(x)$ is a quadratic function.

The five unknown parameters are determined by the four function values from step 1) and one additional function value.

3) Find the minimum of $h(\lambda,\mu)$. For quadratic functions this defines the new point x_{k+1} , for non-quadratic functions x_{k+1} is the point, which corresponds to the minimum of the function values encountered so far. Note that the strategy as given above does not always guarantee such a point and therefore must be modified for the non-quadratic case. We observe that accurate line searches in step 1) are not necessary for step 3), except that they may yield a better approximation for the non-quadratic case.

4) For the next iteration the directions $d_i^{(k)}$, $d_j^{(k)}$ are replaced by the eigenvectors of the quadratic form $h(\lambda,\mu)$:

$$d_i^{(k+1)} = \cos\theta\, d_i^{(k)} - \sin\theta\, d_j^{(k)}$$
$$d_j^{(k+1)} = \sin\theta\, d_i^{(k)} + \cos\theta\, d_j^{(k)}$$
$$d_\ell^{(k+1)} = d_\ell^{(k)} , \qquad \ell \neq i,j ,$$

and:

$$\tan 2\theta = -\frac{2b}{(a-c)} , \quad |\theta| \leq \frac{\pi}{4} .$$

The new set of directions is again orthonormal.

5) For the selection of new pairs of directions Brodlie suggests the following procedure: Choose cyclicly all different pairs of subscripts i,j , therefore $\frac{1}{2}n\,(n-1)$ iterations define one cycle. This is achieved by dividing one cycle into $(n-1)$ blocks of $\frac{1}{2}n$ iterations, and each subscript occurs exactly once in each block. Thus an optimal spread of all directions is attempted to avoid problems as with Smith's method.

Remarks

i) The algorithm maintains orthonormality of the search directions, therefore no problems with linear dependence can occur.

ii) The algorithm does not require accurate line searches.

iii) Conjugacy is achieved only after infinitely many iterations.

4.7. Some numerical results

The following codes for derivative-free algorithms were available:

1) VAO4AD , from Harwell Subroutine Library (HSL): Powell's second procedure |39|.

2) DFMND , from IBM Subroutine Library SLMATH: Zangwill's modification of Powell's first and second procedure |55|.

3) ZXPOW , from International Mathematical and Statistical Libraries (IMSL). Another version of Powell's second procedure. This routine is not available from recent editions of IMSL. The one used in our tests dated back to 1971.

4) PRAXIS , the "PRincipal AXIS" method of Brent | 8 |, published in ALGOL W. We used a FORTRAN translation |29|.

All routines were run on the IBM/37o-168-OS of Kernforschungsanlage Jülich using double precision (64 bits for REAL variables or about 16 decimal places in accuracy). The FORTRAN-H-Extended compiler FORTE was used for compilation.
Test examples were chosen from | 2 |: Problems 3.1 to 3.6 which are six nonlinear least-squares problems fitting one to six Gaussians. The dimension varies from $n=3$ to $n=18$, the degree of difficulty increases rapidly with n : Problem 3.1 is fairly easy, problems 3.5 and 3.6 are extremely hard to solve, and only the best variable metric and Marquardt type methods were found to be successful (see chapter 8, and |2|).

Two criteria for evaluation were used:

I) The final function value after a fixed number of function evaluations (FE), here after 2oo FE. This number is usually exceeded to allow an iteration to be completed.

II) The number of FE to reach a prescribed accuracy ε where

$$f(x) \leq (1+\varepsilon)\, f(x^*) , \qquad \varepsilon = 1o^{-2},\ 1o^{-6},\ 1o^{-1o} .$$

If the number of FE exceeded 2ooo the search was stopped and the final function value recorded.

Table 4.1. Comparison of four derivative-free codes

Probl. ($f(x^*)$)	Code	FE	f_{final} (2oo FE)	T	FE ($1o^{-2}$)	T ($1o^{-2}$)	FE ($1o^{-6}$)	T ($1o^{-6}$)	FE ($1o^{-1o}$)	T ($1o^{-1o}$)
3.1	VAO4AD	2o9	49.•••	o.65	26	o.o8	37	o.12	47	o.15
(49.61...)	DFMND	172	49.•••	o.52	26	o.o9	54	o.17	69	o.21
n=3	ZXPOW	215	49.•••	o.55	26	o.o7	6o	o.15	78	o.21
	PRAXIS				48	o.13	91	o.24	14o	o.36
3.2	VAO4AD	2o2	54.39	o.98	14o	o.68	215	1.o5	254	1.23
(54.39...)	DFMND	215	54.45	1.o3	197	o.95	3o2	1.47	348	1.7o
n=6	ZXPOW	2o1	61.37	o.94	312	1.42	48o	2.28	54o	2.5o
	PRAXIS	>2oo	64.29	o.97	479	2.14	798	3.58	943	4.32
3.3	VAO4AD	219	51.o	1.44	338	2.22	51o	3.36	611	4.o3
(5o.o5...)	DFMND	221	61.58	1.43	417	2.76	667	4.33	814	5.4o
n=9	ZXPOW	232	65.11	1.55	92o	6.21	1334	8.93	156o	1o.42
	PRAXIS	>2oo	58.69	1.42	752	4.85	1385	8.97	5o.o5o3658 [a]	
3.4	VAO4AD	223	72.9	1.85	8o1	6.65	1168	9.7o	13o7	1o.83
(47.92...)	DFMND	227	9o.84	1.86	1o6o	9.18	1759	14.57	2oo1	16.6o
n=12	ZXPOW	225	1oo.32	1.95	1o58	9.o5	2o4o	17.86	47.9255 [a]	
	PRAXIS	>2oo	56.39	1.75	1oo8	8.42	47.9299 [a]		-	
3.5	VAO4AD	236	83.5	2.37	1887	18.86	42.8o5 [a]		-	
(42.4o...)	DFMND	239	89.51	2.39	45.o1 [a]		-		-	
n=15	ZXPOW	233	97.86	2.45	46.17		-		-	
	PRAXIS	>2oo	98.37	2.28	43.59		-		-	
3.6	VAO4AD	222	117.3	2.62	6o.53		-		-	
(39.27...)	DFMND	216	141.3	2.52	75.9o		-		-	
n=18	ZXPOW	215	15o.6	2.66	74.5o		-		-	
	PRAXIS	>2oo	172.7o	2.71	52.o8		-		-	

(criterion I) (criterion II)

(The numbers of the first column, which are put in brackets, indicate the minimum function value. a) Function value after more than 2ooo FE.)

In our original comparison | 2 | we used the elapsed time T taken before and after a call of a minimization code as a measure of performance for criterion II. The values given in Table 4.1 illustrate that both measures FE and T (in seconds) are in surprisingly good agreement with each other (usually measured times are considered to be inaccurate, moreover, they are machine dependent).

When we compare VA04AD, DFMND and ZXPOW, VA04AD turns out to be optimal in all cases, whereas ZXPOW seems consistently inferior. On the other hand PRAXIS shows a rather poor performance for low dimensions and improves systematically with increasing n. We should add that the original random number generator in PRAXIS has been replaced by the IBM routine RANDU, and that repeated runs with different random numbers *highly influenced* our FORTRAN version of PRAXIS.

If the results of Table 4.1 are compared with those in | 2 |, where very powerful algorithms have also been included in the comparison, we conclude that the four derivative-free algorithms behave very similarly in their moderate, but reliable performance. This result is not surprising as the derivative-free methods attempt to find the minimum with a rather limited amount of information, and for higher dimensional problems conjugacy alone is not sufficient to achieve a good performance.

5. Gradient methods

Until now methods were discussed which are entirely derivative-free. It seems, however, most natural to make use of gradients when minimizing a function mainly because a) a vanishing gradient is the necessary condition for a local minimum, b) the negative gradient is the direction of steepest descent which , at least locally, is the best search direction yielding the maximum decrease in function value. In fact the first minimization method was the method of steepest descent proposed by Cauchy in 1847 | 12 |, a convergence proof was given by Goldstein |22| more than 1oo years later.

Besides using gradients directly one may also apply them to build up conjugate directions. A class of methods based on this concept is known as the method of *conjugate gradients* described in more detail in paragraph 5.2.

The most important class of methods uses *gradient differences* to build up approximations to the matrix of second derivatives. These *variable metric methods* work also with conjugate directions and will be presented in great detail in the following chapters.

We begin this chapter with the method of steepest descent and study its behaviour when applied to quadratic functions. The main result will be a new derivation of some interesting properties first given by Akaike | 1 |.

5.1. The method of steepest descent

Whenever the gradient $g := \nabla f$ [1] of the function $f : \mathbb{R}^n \to \mathbb{R}$ is known, a straightforward algorithm would be to use the search direction defined at x_k by

$$p_k := -g_k \,, \tag{5.1}$$

where g_k is the gradient at x_k . Moreover a strategy to find a adequate step-length along p_k will be needed. In the case

---

[1] where $\nabla := (\frac{\partial}{\partial \xi_1}, \ldots, \frac{\partial}{\partial \xi_n})^T$

that an exact line search is performed, this variant of the method of steepest descent is known as *optimum gradient method*. For the rest of this paragraph we limit the discussion to this special case and furthermore study only quadratic , positive definite, functions

$$f(x) = \frac{1}{2}\, x^T Gx + b^T x + c \qquad (5.2a)$$

$$= \frac{1}{2}\, (x - x^*)^T G\, (x - x^*) + f(x^*)\ , \qquad (5.2b)$$

with x^* denoting the minimum of $f(x)$. Obviously

$$g_k = G\, (x_k - x^*)\ .$$

We want to derive an expression for the optimal step-length. With

$$h(\lambda) := f(x_k + \lambda p_k) = f(x_k - \lambda g_k)$$

$$= \frac{1}{2}\, \lambda^2\, g_k^T Gg_k - \lambda g_k^T G(x_k - x^*) + f(x_k) \qquad (5.3)$$

we obtain from $h'(\lambda) = o$ the optimal step-length $\lambda = \alpha_k$:

$$\alpha_k = \frac{g_k^T G(x_k - x^*)}{g_k^T Gg_k} = \frac{g_k^T g_k}{g_k^T Gg_k}\ . \qquad (5.4)$$

For later purposes we need an expression for the reduction in function value, when stepping from x_k to $x_{k+1} = x_k - \alpha_k g_k$. Inserting $\lambda = \alpha_k$ from (5.4) into (5.3) we find

$$f(x_{k+1}) = f(x_k) - \frac{1}{2}\, \frac{(g_k^T g_k)^2}{g_k^T Gg_k}\ ,$$

or

$$f(x_{k+1}) - f(x^*) = R_k\, \{f(x_k) - f(x^*)\}\ . \qquad (5.5)$$

Using (5.2b), R_k can be expressed as

$$R_k := 1 - D_k = 1 - \frac{(g_k^T g_k)^2}{(g_k^T Gg_k)\, g_k^T\, (x_k - x^*)}\ . \qquad (5.6)$$

Oren and Luenberger |36| introduced the name "single-step convergence rate" for R_k as defined by (5.5). With the inequalities

$$f(x_k) > f(x_{k+1}) > f(x^*)$$

we deduce from (5.5) that $o < R_k < 1$. In particular for the method of steepest descent, $R_k > o$ means (see 5.6) that

$$(g_k^T g_k)^2 < (g_k^T G g_k)(g_k^T G^{-1} g_k) ,$$

or that the Schwarz inequality holds. For $G = cI$, this inequality degenerates to an equality, or $R_o = o$, which expresses the fact that quadratic functions with spherical contour lines are minimized in one step by the optimum gradient method.

5.1.1. An example

Given $f(\xi,\eta) = \xi^2 + \eta^2 - \xi\eta - 2\xi - \eta$, and $x_o = (1,o)^T$.

The optimum gradient method generates the following sequence for x_k and g_k (only the first 5 members are given):

$$x_k : \begin{pmatrix}1\\o\end{pmatrix} , \begin{pmatrix}1\\1\end{pmatrix} , \frac{1}{2}\begin{pmatrix}3\\2\end{pmatrix} , \frac{1}{4}\begin{pmatrix}6\\5\end{pmatrix} , \frac{1}{8}\begin{pmatrix}13\\10\end{pmatrix} ,$$

$$g_k : -\begin{pmatrix}o\\2\end{pmatrix} , -\begin{pmatrix}1\\o\end{pmatrix} , -\frac{1}{2}\begin{pmatrix}o\\1\end{pmatrix} , -\frac{1}{4}\begin{pmatrix}1\\o\end{pmatrix} , -\frac{1}{8}\begin{pmatrix}o\\1\end{pmatrix} .$$

The general law can be expressed using $x^* = \frac{1}{3}(5,4)^T$:

$$x_{2k} = x^* - \frac{1}{3\cdot 2^{2k-1}}\begin{pmatrix}1\\2\end{pmatrix} , \quad x_{2k+1} = x^* - \frac{1}{3\cdot 2^{2k}}\cdot\begin{pmatrix}2\\1\end{pmatrix} ,$$

$$g_{2k} = -\frac{1}{2^{2k-1}}\begin{pmatrix}o\\1\end{pmatrix} , \quad g_{2k+1} = -\frac{1}{2^{2k}}\cdot\begin{pmatrix}1\\o\end{pmatrix} .$$

Fig. 5.1 illustrates this sequence.

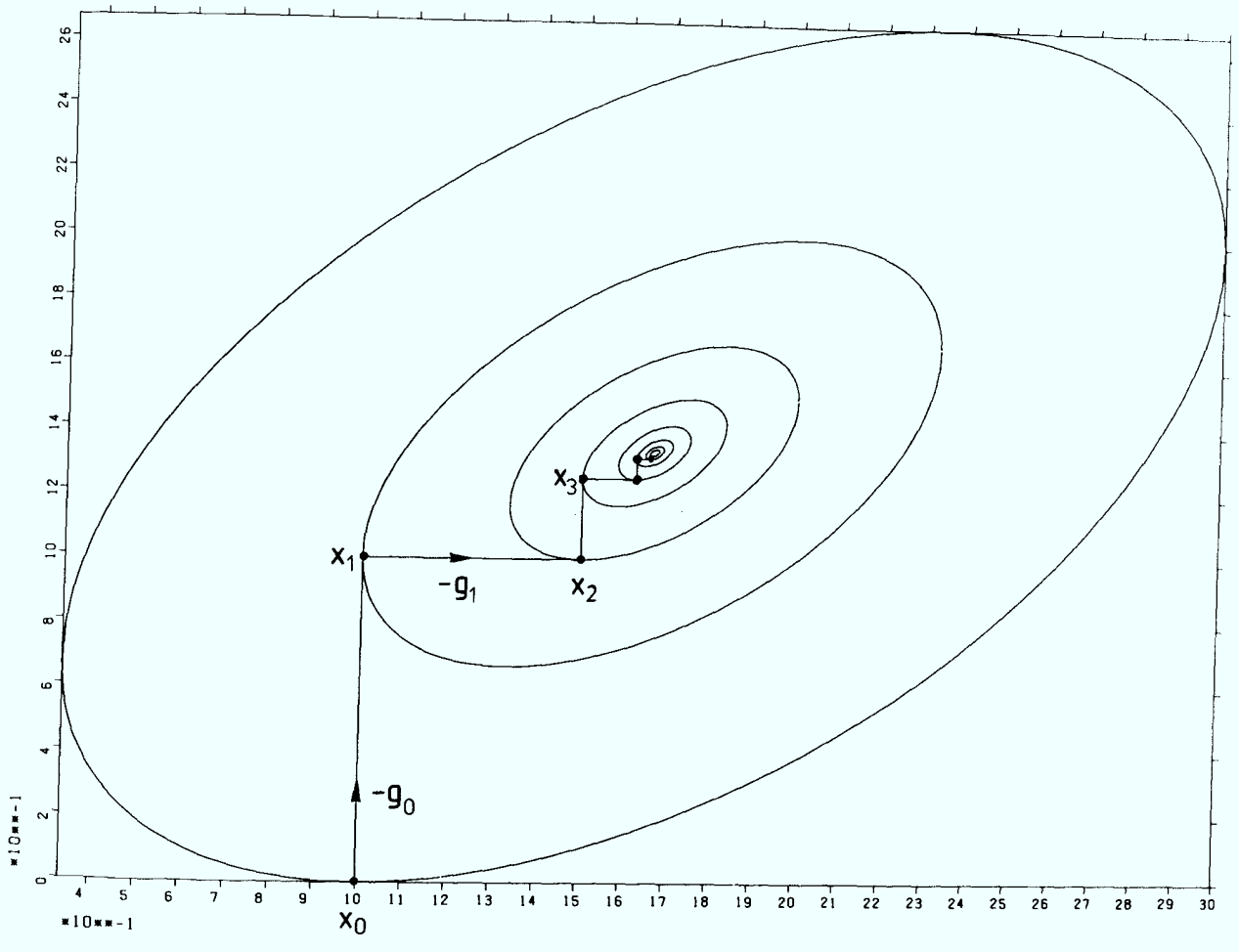


Fig. 5.1: Optimum gradient method for n=2 .

This example already demonstrates that the method of steepest descent does not possess the property of quadratic termination, instead a convergence rate according to a geometrical progression is revealed in this example. These properties will be studied in more detail, but as already stated at the beginning, only quadratic functions will be considered.

5.1.2. The single-step convergence rate

In this section we show that R_k is in general *monotonously increasing*, or D_k, defined in (5.6), decreasing with k, respectively.

For the following it is most convenient to decompose $(x_k - x^*)$ with respect to the eigenvectors v_i of G. As G was assumed to be positive definite, its eigenvalues λ_i are positive and its eigenvectors v_i orthogonal, further $v_i^T v_i = 1$ will be assumed. We have

$$x_k - x^* = \sum_{i=1}^{n} c_{k,i} v_i , \tag{5.7a}$$

$$g_k = G(x_k - x^*) = \sum_{i=1}^{n} c_{k,i} \lambda_i v_i ,$$

$$x_{k+1} - x^* = x_k - x^* - \alpha_k g_k$$

$$= \sum_{i=1}^{n} c_{k,i} (1 - \alpha_k \lambda_i) v_i . \tag{5.7b}$$

Therefore the following nonlinear recurrence relations hold:

$$c_{k+1,i} = (1 - \alpha_k \lambda_i) c_{k,i} \tag{5.8}$$

with α_k from (5.4):

$$\alpha_k = \frac{\sum_{i=1}^{n} c_{k,i}^2 \lambda_i^2}{\sum_{i=1}^{n} c_{k,i}^2 \lambda_i^3} . \tag{5.9}$$

For the rest we use the abbreviations

$$w_i := c_{k,i}^2 , \quad b_\ell := \sum_{i=1}^{n} c_{k,i}^2 \lambda_i^\ell > 0 .$$

Then from (5.6):

$$D_k = \frac{(\sum_{i=1}^{n} w_i \lambda_i^2)^2}{(\sum_{i=1}^{n} w_i \lambda_i)(\sum_{i=1}^{n} w_i \lambda_i^3)} = \frac{b_2^2}{b_1 b_3} ,$$

and

$$D_{k+1} = \frac{\{\sum_{i=1}^{n} w_i(1 - \alpha_k\lambda_i)^2 \lambda_i^2\}^2}{\{\sum_{i=1}^{n} w_i(1 - \alpha_k\lambda_i)^2 \lambda_i\}\{\sum_{i=1}^{n} w_i(1 - \alpha_k\lambda_i)^2 \lambda_i^3\}}$$

$$= \frac{b_2^2(-b_3^2 + b_2b_4)^2}{b_3(b_1b_3 - b_2^2)(b_3^3 - 2b_2b_3b_4 + b_5b_2^2)} \quad .$$

Comparing the expressions for D_k and D_{k+1}, the condition $D_{k+1} \leq D_k$ amounts to showing that

$$r_k := b_5(b_1b_3 - b_2^2) + b_3(b_2b_4 - b_3^2) - b_4(b_1b_4 - b_2b_3) \geq 0 \ .$$

The three terms in brackets can be expressed more conveniently. Firstly, we have

$$b_1b_3 - b_2^2 = \sum w_iw_j(\lambda_i\lambda_j^3 - \lambda_i^2\lambda_j^2)$$

$$= \sum_{i<j} w_iw_j\lambda_i\lambda_j^2(\lambda_j - \lambda_i) + \sum_{i>j} w_iw_j\lambda_i\lambda_j^2(\lambda_j - \lambda_i)$$

$$= \sum_{i<j} w_iw_j\lambda_i\lambda_j\{\lambda_j(\lambda_j - \lambda_i) - \lambda_i(\lambda_j - \lambda_i)\}$$

$$= \sum_{i<j} w_iw_j\lambda_i\lambda_j(\lambda_j - \lambda_i)^2 \ .$$

Similarly, we have

$$b_2b_4 - b_3^2 = \sum_{i<j} w_iw_j\lambda_i^2\lambda_j^2(\lambda_j - \lambda_i)^2 \ ,$$

$$b_1b_4 - b_2b_3 = \sum_{i<j} w_iw_j\lambda_i\lambda_j(\lambda_i + \lambda_j)(\lambda_j - \lambda_i)^2 \ .$$

Finally, we obtain

$$r_k = \sum_{\ell=1}^{n} \sum_{i<j} w_iw_jw_\ell\lambda_i\lambda_j(\lambda_j - \lambda_i)^2\{\lambda_\ell^5 + \lambda_\ell^3\lambda_i\lambda_j - \lambda_\ell^4(\lambda_i + \lambda_j)\}$$

$$= \sum_{\ell=1}^{n} \sum_{i<j} w_iw_jw_\ell\lambda_i\lambda_j\lambda_\ell^3(\lambda_j - \lambda_i)^2(\lambda_\ell - \lambda_i)(\lambda_\ell - \lambda_j) \ . \qquad (5.10)$$

For $n \geq 3$ this can be written as

$$r_k = \sum_{\ell<i<j} w_i w_j w_\ell \lambda_i \lambda_j \lambda_\ell (\lambda_i - \lambda_j)^2 (\lambda_j - \lambda_\ell)^2 (\lambda_\ell - \lambda_i)^2 \geq o \ , \quad (5.11)$$

which is the desired result proving $R_{k+1} \geq R_k$.

Three possible realizations of the set of eigenvalues will be discussed in more detail.

a) If all λ_i are the same, $\lambda_i = \lambda_1$ say, relation (5.9) yields $\alpha_k = 1/\lambda_1$, and therefore from (5.8):

$$c_{k+1,i} = o \ , \quad i = 1(1)n \ , \quad k \geq o \ ,$$

and (5.7) reduces to the simple result mentioned before: $x_1 = x^*$.

b) If only *two eigenvalues* are different, which also applies quite generally to the case $n=2$, equ. (5.1o) leads to the interesting result that $r_k = o$, or $R_{k+1} = R_k = \text{const.}$

c) If at least *three eigenvalues* are different, and if further the corresponding eigenvector components $c_{k,i}$ do not vanish, equ. (5.11) leads to the stronger result $R_{k+1} > R_k$. In other words:

> The single-step convergence rate of the optimum gradient method applied to quadratic functions with more than two variables deteriorates in general with each iteration.

We shall see later that $c_{k,i} \neq o$ if $c_{o,i} \neq o$, whereas from (5.8) we have $c_{k,i} = o$ if $c_{o,i} = o$.

5.1.3. Upper bound for the single-step convergence rate

In the previous section we showed that R_k is in general monotonously increasing. We now derive the "classical" upper bound for R_k which, however, will usually not be achieved (see the next section). The following derivation is mainly based on the one given by Kowalik and Osborne |28|, pp. 34. For the rest we assume

that $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.

The definition of a perfect line search along $-g_k$ with x_{k+1} as the minimum implies the inequality

$$f(x_{k+1}) \leq f(x_k - \lambda g_k) \quad , \quad \text{for any} \quad \lambda \geq o \; . \tag{5.12}$$

Next we apply the eigenvalue decomposition analogous to (5.7b) to the right hand side of (5.12) by replacing α_k by λ in (5.7b):

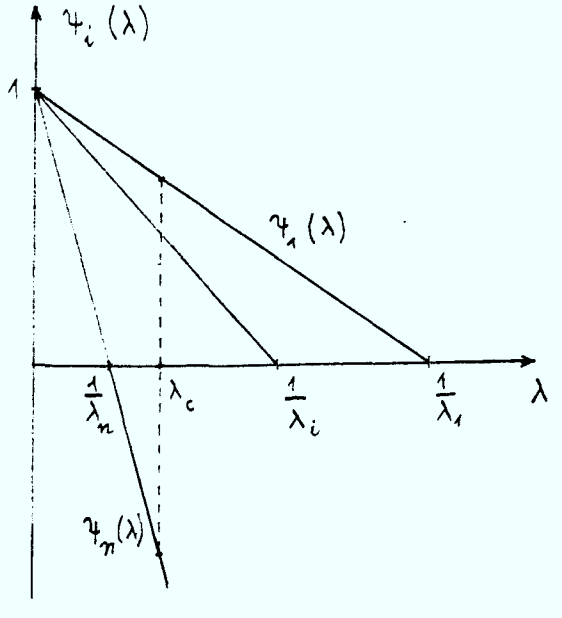
$$\begin{aligned} f(x_k - \lambda g_k) &= \frac{1}{2} (x_k - x^* - \lambda g_k)^T G(x_k - x^* - \lambda g_k) + f(x^*) \\ &= \frac{1}{2} \sum_{i=1}^{n} c_{k,i}^2 (1 - \lambda\lambda_i)^2 \lambda_i + f(x^*) \; . \end{aligned} \tag{5.13}$$

Given the set of parabolas

$$\chi_i(\lambda) = (1 - \lambda\lambda_i)^2 \quad , \quad i = 1(1)n \; .$$

We want to solve the problem

$$\min_{\lambda > o} \; \max_{1 \leq i \leq n} \chi_i(\lambda) \; . \tag{5.14}$$



Consider the set of straight lines

$$\psi_i(\lambda) = 1 - \lambda\lambda_i \quad , \quad i = 1(1)n \; .$$

As becomes clear from the figure, for any $\lambda > o$ $\max\limits_{1 \leq i \leq n} |\psi_i(\lambda)|$ will be assumed either for $i=1$ or $i=n$, depending on the condition

$$\psi_1(\lambda) \geq |\psi_n(\lambda)|$$

for the positive range of $\psi_1(\lambda)$.

From $\psi_1(\lambda_c) = -\psi_n(\lambda_c)$ we obtain

$$\lambda_c = \frac{2}{(\lambda_1 + \lambda_n)} \tag{5.15}$$

and therefore for $\lambda > o$:

$$\max_{1\leq i\leq n} |\psi_i(\lambda)| = \begin{cases} 1 - \lambda\lambda_1 , & \lambda \leq \lambda_c , \\ \lambda\lambda_n - 1 , & \lambda > \lambda_c . \end{cases}$$

Moreover:

$$\min_{\lambda>o} \max_{1\leq i\leq n} |\psi_i(\lambda)| = 1 - \lambda_c\lambda_1 = \lambda_c\lambda_n - 1 = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} .$$

The solution of (5.14) follows immediately:

$$\max_{1\leq i\leq n} \chi_i(\lambda_c) = \min_{\lambda>o} \max_{1\leq i\leq n} \chi_i(\lambda) = \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 . \tag{5.16}$$

Since (5.12) is valid for *any* $\lambda>o$, it is in particular valid for $\lambda=\lambda_c$ with λ_c from (5.15). Together with (5.13) we obtain the inequality

$$f(x_{k+1}) - f(x^*) \leq \frac{1}{2} \sum_{i=1}^{n} c_{k,i}^2 \chi_i(\lambda_c)\lambda_i .$$

With (5.16) we can give an upper bound for $\chi_i(\lambda_c)$:

$$\chi_i(\lambda_c) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 , \quad \text{for all } i .$$

With

$$f(x_k) - f(x^*) = \frac{1}{2} \sum_{i=1}^{n} c_{k,i}^2 \lambda_i$$

we arrive at the final result

$$f(x_{k+1}) - f(x^*) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \{f(x_k) - f(x^*)\}. \tag{5.17}$$

Inequality (5.17) is a special case of the *Kantorovich inequality* (see e.g. Luenberger |31|, pp. 151).

Let us consider two extreme cases.

a) $\lambda_n = \lambda_1$: All eigenvalues are equal. As a consequence of (5.17), the optimum gradient method will find the minimum of an n-dimensional sphere in one step.

b) $\lambda_n >> \lambda_1$: For highly ill-conditioned matrices the upper bound of the single-step convergence rate R_k will be very close to one.

This latter statement does not imply that the upper bound (5.17) will really be achieved. Take for instance the case $n=2$. We saw in section 5.1.2. that for $n=2$ the convergence rate is strictly geometric and depends only on the choice of the initial point x_o . In this case the constant value of R_k can be much smaller than its upper bound given by (5.17).

On the other hand, the numerical examples of section 5.1.5. will reveal that for $n \geq 3$ and very *ill-conditioned problems* this upper bound can be reached very closely. In section 5.1.2. it was shown that for $n \geq 3$ R_k is in general strictly monotonously increasing. Akaike | 1 | succeeded in proving that the asymptotic limit of R_k is in fact very close to the upper bound (5.17). Therefore, for $n \geq 3$, *the optimum gradient method is most unsuitable for ill-conditioned problems.* (Note that Akaike assumes implicitly the existence of at least *three different eigenvalues* (p. 12: "under the condition which assumes that the point λ_i is not discarded...") thus excluding the special behaviour for the case of only two different eigenvalues.)

It was our purpose to present these details, because the literature is somewhat misleading in what concerns the Akaike results. Kowalik and Osborne |28|, p. 37, after having derived inequality (5.17), write:

> "Actually the rate of convergence is exactly geometric, and a proof of the complementary inequality is given in Forsythe."

We were unfortunately not able to get hold of the Forsythe report. More recently, Luenberger |31|, p. 153, stated:

"It also has been shown, by Akaike, that barring certain degenerate starting points, this bound on the rate of convergence is exact. The proof of this fact is fairly complex and we do not give it here; but because of that fact, we say that the convergence ratio of steepest descent *is* $(\lambda_n - \lambda_1)^2/(\lambda_n + \lambda_1)^2$."

However, on p. 167, Luenberger weakens this statement saying:

"For a proof that the estimate (5.17) is *essentially* exact see Akaike."

5.1.4. The asymptotic behaviour of the optimum gradient method

In this section we present a new derivation of a result given by Akaike | 1 |: Asymptotically the optimum gradient method is entirely restricted to the search along *two* directions. Although the derivation to be presented could be formulated more rigorously, it is our main purpose to give a better insight into the behaviour of this method than the rather difficult presentation of Akaike which is based on arguments of probability theory.

Our derivation is completely concerned with the study of the *nonlinear recurrence relations* (5.8) for the components of the eigenvalue decomposition (5.7):

$$c_{k+1,i} = a_{k,i}\, c_{k,i} \quad , \quad i = 1(1)n \ , \quad k = o,1,\ldots, \qquad (5.18a)$$

with

$$a_{k,i} := (1 - \alpha_k \lambda_i) \ . \qquad (5.18b)$$

We shall assume that all eigenvalues are different: $\lambda_1 < \lambda_2 < \ldots < \lambda_n$, and that $c_{o,i} \neq o$, $i = 1(1)n$ (if $c_{o,i} = o$ for some i , the problem would be reduced to a lower-dimensional one). Then the following three properties hold for any k :

a) $$a_{k,1} > a_{k,2} > \ldots > a_{k,n} \ . \qquad (5.19a)$$

This follows immediately from $\alpha_k > o$ according to (5.9) and the fact that the λ_i are ordered.

b) $o < a_{k,1} < 1$. (5.19b)

By definition (5.18b): $a_{k,1} < 1$. Furthermore:

$$a_{k,1} = \frac{1}{\sum_i c_{k,i}^2 \lambda_i^3} \sum_{i=2}^{n} c_{k,i}^2 \lambda_i^2 (\lambda_i - \lambda_1) > o .$$

c) $a_{k,n} < o$. (5.19c)

This follows from

$$a_{k,n} = \frac{-1}{\sum_i c_{k,i}^2 \lambda_i^3} \sum_{i=1}^{n-1} c_{k,i}^2 \lambda_i^2 (\lambda_n - \lambda_i) < o .$$

We shall further need the formal solution of (5.18a) (the α_k depend also on $c_{k,i}$):

$$c_{k+1,i} = a_{k,i} a_{k-1,i} \cdots a_{o,i} c_{o,i} . \tag{5.2o}$$

From $a_{k,1} > o$ and $a_{k,n} < o$ we obtain for α_k the bounds

$$\frac{1}{\lambda_n} < \alpha_k < \frac{1}{\lambda_1} , \tag{5.21}$$

thus the sequence of α_k has at least one accumulation point. We want to investigate the implications which result from two different assumptions concerning the limiting behaviour of α_k .

i) *The sequence of* α_k *is convergent.*

Let α denote the limiting value of α_k . Then we obtain from (5.2o) the asymptotic behaviour of $c_{k,i}$:

$$c_{k,i} \cong \text{const}(1 - \alpha\lambda_i)^k , \quad k \geq k_o , \tag{5.22}$$

where const depends on i , but not on k .

Three different assumptions for the value of α will be discussed now.

1) $$1 - \alpha\lambda_1 < \alpha\lambda_n - 1 \ . \tag{5.23}$$

This implies that with (5.22) and (5.19a) we have

$$\lim_{k\to\infty} \frac{c_{k,i}}{c_{k,n}} = o \ , \qquad i = 1(1)n-1 \ . \tag{5.24}$$

On the other hand, using the definition of α_k , (5.24) would lead to the asymptotic value

$$\alpha_k := \frac{\sum_{i=1}^{n} c_{k,i}^2 \lambda_i^2}{\sum_{i=1}^{n} c_{k,i}^2 \lambda_i^3} \cong \frac{1}{\lambda_n} \ .$$

This, however, is in contradiction to (5.23).

2) Similarly, the assumption $(1 - \alpha\lambda_1) > (\alpha\lambda_n - 1)$ leads to a contradiction. It remains to consider the equal sign.

3) We assume therefore that $\alpha = 2/(\lambda_1 + \lambda_n)$. Instead of (5.24) we have now

$$\lim_{k\to\infty} \frac{c_{k,i}}{c_{k,n}} = o \qquad i = 2(1)n-1 \ . \tag{5.25}$$

Let

$$r_k := \frac{c_{k,1}}{c_{k,n}} \ , \tag{5.26}$$

then (5.2o) leads to the recurrence relation

$$r_{k+1} = \frac{(1 - \alpha_k\lambda_1)}{(1 - \alpha_k\lambda_n)} r_k \ , \qquad k = o,1,\ldots, \tag{5.27}$$

and with $(1 - \alpha\lambda_1) = (\alpha\lambda_n - 1)$ asymptotically to

$$r_{k+1} \cong (-1)^k r_k \ , \qquad k \geq k_o \ . \tag{5.28}$$

Let ρ denote the limiting value of r_k^2. Then, using (5.25), the following relation between ρ and α holds:

$$\alpha = \frac{\rho\,\lambda_1^2 + \lambda_n^2}{\rho\,\lambda_1^3 + \lambda_n^3},$$

or $\rho = (\lambda_n/\lambda_1)^2$. Thus in this case no contradiction occurs.

Consider the single-step convergence rate $R_k = 1 - D_k$. In section 5.1.2. it was shown that the sequence

$$D_k = \alpha_k \beta_k$$

with

$$\beta_k := \frac{\sum_{i=1}^{n} c_{k,i}^2\,\lambda_i^2}{\sum_{i=1}^{n} c_{k,i}^2\,\lambda_i}$$

is monotonously decreasing, thus the product $\alpha_k\beta_k$ is *always convergent*. Using again (5.25), we obtain with ρ as given above:

$$\beta = \frac{\rho\,\lambda_1^2 + \lambda_n^2}{\rho\,\lambda_1 + \lambda_n} = \frac{2\lambda_1\lambda_n}{\lambda_1 + \lambda_n},$$

or

$$R = \frac{(\lambda_n - \lambda_1)^2}{(\lambda_n + \lambda_1)^2}.$$

In other words: If the initial point x_o is chosen [1] such that α_k converges to the value

$$\alpha = \frac{2}{\lambda_1 + \lambda_n},$$

---

1) For $n=2$ such a choice is always possible (see the end of this section), for $n \geq 3$ it would be interesting to know if such points do exist.

the optimum gradient method will perform in its worst possible way, because the single-step convergence rate will converge to its upper bound (see section 5.1.3.).

ii) *The sequence of* α_k *has two limiting values.*

We know from section 5.1.3. that in general, i.e. for arbitrary starting points x_o, the single-step convergence rate will not achieve its upper bound, therefore α_k will in general not converge to $\alpha = 2/(\lambda_1 + \lambda_n)$. As a consequence of the preceding discussion, we conclude that the sequence of α_k cannot have one limiting value in this case.

From (5.19c) we have $\alpha_{k,n} < o$, therefore $c_{k,n}$ is, according to (5.2o), strictly oscillating with k, the same is true for the ratio r_k defined by (5.26). Let us assume that r_k does not tend to zero. Then we can expect, that in general $|r_{2k}|$ and $|r_{2k+1}|$ will tend to different limits.

These plausibility arguments only serve to try the two limiting values for the sequence of α_k :

$$\alpha := \lim_{k\to\infty} \alpha_{2k} \quad , \quad \alpha' := \lim_{k\to\infty} \alpha_{2k+1} \; .$$

Then (5.22) is generalized to the asymptotic behaviour

$$c_{k,i} \cong \text{const } (1 - \alpha\lambda_i)^{k/2} \; (1 - \alpha'\lambda_i)^{k/2} \quad , \quad k \geq k_o \; . \qquad (5.22')$$

The following considerations are completely analogous to those when one limiting value was assumed.

The assumption

$$(1 - \alpha\lambda_1)(1 - \alpha'\lambda_1) < (1 - \alpha\lambda_n)(1 - \alpha'\lambda_n) \qquad (5.23')$$

leads again to (5.24) or, with the definition of α_k, to $\alpha = \alpha' = 1/\lambda_n$. The contradiction follows immediately from (5.23'). As before, only the equal sign does not lead to a contradiction.

From

$$(1 - \alpha\lambda_1)(1 - \alpha'\lambda_1) = (1 - \alpha\lambda_n)(1 - \alpha'\lambda_n) \tag{5.29}$$

it follows that the two limiting values α,α' are related by

$$\frac{1}{\alpha} + \frac{1}{\alpha'} = \lambda_1 + \lambda_n \ .$$

Again, the equality condition (5.29) implies that (5.25) holds. Therefore, with r_k defined by (5.26), we obtain the asymptotic relation for α_k :

$$\alpha_k \cong \frac{r_k^2 \lambda_1^2 + \lambda_n^2}{r_k^2 \lambda_1^3 + \lambda_n^3} \ . \tag{5.3o}$$

This inserted into (5.27) yields the asymptotic recurrence relation for r_k :

$$r_{k+1} \cong - \left(\frac{\lambda_n}{\lambda_1}\right)^2 \frac{1}{r_k} \ , \quad k \geq k_o \ .$$

The solution is

$$r_k = \begin{cases} r_{k_o} \ , & k = k_o+2m \ , \quad m = o,1,\ldots \\ - \left(\dfrac{\lambda_n}{\lambda_1}\right)^2 \dfrac{1}{r_{k_o}} \ , & k = k_o+2m+1 \ , \quad m = o,1,\ldots \end{cases} \tag{5.31}$$

With (5.31) and (5.25) we have found Akaike's main result: When applied to quadratic functions, the optimum gradient method will approach the minimum asymptotically by searching along two directions only. These directions lie in the plane spanned by the two eigenvectors of G , which correspond to the smallest and the largest eigenvalue.

Next we want to give three equivalent expressions for the asymptotic value of the single-step convergence rate R_k . Because of the monotonicity of R_k we consider only the limit of R_{2k} . Then we have

$$R_{2k} = 1 - \alpha_{2k}\,\beta_{2k}$$

$$\cong 1 - \alpha\,\frac{r_{2k}^2\,\lambda_1^2 + \lambda_n^2}{r_{2k}^2\,\lambda_1 + \lambda_n} \quad .$$

Expressing r_{2k}^2 by its asymptotic value, which follows from (5.3o), we obtain the asymptotic single-step convergence rate

$$R(\alpha) = \frac{(1 - \alpha\lambda_1)(\alpha\lambda_n - 1)}{\alpha(\lambda_1 + \lambda_n) - 1} \quad . \tag{5.32}$$

It is easily verified that $R(\alpha)$ assumes its maximum value for $\alpha = 2/(\lambda_1 + \lambda_n)$, where R coincides with the upper bound (5.17).

A more symmetric expression for R follows, if we introduce a quantity ε which is a measure for the deviation from the worst possible case:

$$\varepsilon := \lambda_1 + \lambda_n - \frac{2}{\alpha} \quad .$$

Then, according to (5.21), ε is confined to

$$-(\lambda_n - \lambda_1) \leq \varepsilon \leq (\lambda_n - \lambda_1) \quad ,$$

and (5.32) becomes

$$R(\varepsilon) = \frac{(\lambda_n - \lambda_1)^2 - \varepsilon^2}{(\lambda_n + \lambda_1)^2 - \varepsilon^2} \quad . \tag{5.32'}$$

If we set

$$\varepsilon^2 = \frac{(\lambda_n - \lambda_1)^2\,(c - \frac{1}{c})^2}{4 + (c - \frac{1}{c})^2} \quad ,$$

(5.32') becomes

$$R(c) = \frac{(\lambda_n - \lambda_1)^2}{(\lambda_n + \lambda_1)^2 + \lambda_n \lambda_1 (c - \frac{1}{c})^2} , \qquad (5.32")$$

which is Akaike's expression for R (p. 11 of | 1 |).

We conclude this section considering the special case n=2 . Then (5.31) is *strictly valid for* $k_o = o$:

$$r_{2k} = r_o , \quad r_{2k+1} = -\left(\frac{\lambda_2}{\lambda_1}\right)^2 \frac{1}{r_o} , \quad r_o = \frac{c_{o,1}}{c_{o,2}} .$$

If we select x_o such that $|r_o| = \lambda_2/\lambda_1$, we obtain $r_{k+1} = (-1)^k r_k$, $k \geq o$, i.e. (5.28), and with the (for n=2 exact) formula (5.3o): $\alpha_k = 2/(\lambda_1 + \lambda_2)$. Therefore, the choice

$$\frac{c_{o,1}}{c_{o,2}} = \pm \frac{\lambda_2}{\lambda_1}$$

leads to the slowest possible single-step convergence rate.

Consider again example 5.1.1. With

$$G = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} , \quad \lambda_1 = 1 , \ \lambda_2 = 3 , \quad v_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} , \ v_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

and $x_o = (1,o)^T$, $x^* = \frac{1}{3}(5,4)^T$ one confirms easily that $c_{o,1} = -\sqrt{2}$, $c_{o,2} = \sqrt{2}/3$, or $r_o = -\lambda_2/\lambda_1$. In example 5.1.1. x_o was in fact chosen in a worst possible manner.

5.1.5. Some numerical results

In order to illustrate the performance of the optimum gradient method we choose the quadratic function

$$f(x) = \frac{1}{2} x^T G x , \quad G_{ik} = \frac{1}{i+k-1} , \quad x_o = (1,1,\ldots,1)^T .$$

G is the Hilbert matrix which is known to become highly ill-conditioned already for small n .

All numerical results were obtained on the IBM/37o-168 in fourfold precision, i.e. with about 32 decimal places in accuracy. Table 5.1 shows results for n=2,3,4,5. Function values are given after $1o^{\ell}$ iterations with ℓ=1,2,...,5. Also given are the euclidean norms $\| x_{fin} \|$, $\| g_{fin} \|$ of x_k , g_k after the final iteration. R is the asymptotic single-step convergence rate (5.32') as obtained from the numerical results after k iterations, and R_{max} its upper bound (for ε=o).

it.	n = 2	n = 3	n = 4	n = 5
1o	o.226•1o^{-23}	o.122•1o^{-4}	o.771•1o^{-4}	o.216•1o^{-3}
1o^2	o.154•1o^{-75} a)	o.596•1o^{-5}	o.15o•1o^{-4}	o.152•1o^{-4}
1o^3		o.452•1o^{-8}	o.191•1o^{-7}	o.126•1o^{-6}
1o^4		o.289•1o^{-39}	o.183•1o^{-8}	o.131•1o^{-9}
1o^5		o.627•1o^{-74} b)	o.117•1o^{-18}	o.154•1o^{-1o}
$\| x_{fin} \|$:	o.447•1o^{-38}	o.215•1o^{-35}	o.491•1o^{-7}	o.3o6•1o^{-2}
$\| g_{fin} \|$:	o.294•1o^{-39}	o.746•1o^{-38}	o.638•1o^{-11}	o.134•1o^{-7}
R :	o.oo425	o.992o5o1o	o.9997392o	o.99999145
	(k $\geq$ 1)	(k $\geq$ 4o)	(k $\geq$ 14oo)	(k $\geq$ 3o ooo)
R_{max}:	o.8125o	o.99239628	o.9997422o	o.99999161

Table 5.1: Results of optimum gradient method applied to Hilbert function. (a): Final function value for k=33 ,
b): for k=2o ooo)

As could be expected, R can be very small for $n=2$ in accordance with the results of section 5.1.2. and the remarks at the end of section 5.1.3. Moreover R is constant in this case thus convergence is achieved in a tolerable number of iterations.

For $n \geq 3$ R is always found very close to the worst possible value R_{max} thereby slowing down the convergence to an unacceptable rate already for $n=3$. As was anticipated at the end of section 5.1.3., a theoretical explanation for this behaviour was given by Akaike.

5.2. The method of conjugate gradients

The method of conjugate gradients was originally designed by Hestenes and Stiefel |23| to solve systems of linear equations iteratively. For instance, consider

$$f(x) = \frac{1}{2} (Ax - b)^T (Ax - b)$$
$$= \frac{1}{2} x^T A^T Ax - b^T Ax + \frac{1}{2} b^T b .$$

For any nonsingular A the matrix $A^T A$ is positive definite and the minimum x^* of $f(x)$ satisfies $Ax^* = b$. If A is known to be positive definite one may also consider (see Stoer and Bulirsch |1o|):

$$f(x) = \frac{1}{2} (Ax - b)^T A^{-1} (Ax - b)$$
$$= \frac{1}{2} x^T Ax - b^T x + \frac{1}{2} b^T A^{-1} b .$$

The minimum of $f(x)$ solves again $Ax^* = b$. For the actual minimization one would of course ignore the constant term which contains A^{-1} .

For the rest of this paragraph we assume $f(x)$ to be a quadratic function

$$f(x) = \frac{1}{2} x^T Gx + b^T x + c$$

with gradient

$$g(x) = Gx + b \ .$$

Let x_k, x_{k+1} be two points with gradients g_k, g_{k+1}, respectively. Then the gradient difference is related to the difference of the positions by

$$g_{k+1} - g_k = G\,(x_{k+1} - x_k) \ . \tag{5.33a}$$

Therefore, any vector p satisfying $p^T (g_{k+1} - g_k) = o$ will be conjugate to $x_{k+1} - x_k$. Equation (5.33a) can be considered as the relation which had *greatest influence in the development of minimization algorithms which use derivatives*. The two differences appearing in (5.33a) will be used continuously in the following chapters and a special notation seems most adequate. We want to adopt the Fletcher notation |2o|:

$$\begin{aligned} s_k &:= x_{k+1} - x_k \ , \\ y_k &:= g_{k+1} - g_k \ . \end{aligned} \tag{5.34}$$

Then (5.33a) becomes

$$y_k = G\, s_k \ . \tag{5.33b}$$

With (5.33) we deduce that the change of *gradients* offers a simple way to construct *conjugate* directions. This led to the method of *conjugate gradients*.

5.2.1. A description of the method

The method of conjugate gradients works as follows. Given a starting point x_o and a search direction which is equal to the direction of steepest descent, $p_o := -g_o = -g(x_o)$. Determine the minimum x_1 along p_o and calculate g_1. We shall assume that the new search direction is a *biased steepest descent* direction of the form

$$p_1 = -g_1 + \beta_{1o}\, p_o \ ,$$

and determine β_{1o} such that p_1 is conjugate to p_o. With $x_1 = x_o + \alpha_o p_o$ it follows immediately that the conjugacy of p_1 and p_o

is equivalent to the orthogonality of p_1 and y_o :

$$p_1^T y_o = p_1^T G s_o = \alpha_o p_1^T G p_o = o .$$

Now:

$$p_1^T y_o = (-g_1 + \beta_{1o} p_o)^T (g_1 - g_o)$$
$$= -g_1^T g_1 + \beta_{1o} g_o^T g_o ,$$

where we used the perfect line search condition $p_o^T g_1 = o$ which with $p_o = -g_o$ also implies $g_o^T g_1 = o$. Therefore, p_1 is conjugate to p_o if $\beta_{1o} = g_1^T g_1 / g_o^T g_o$.

Next we determine x_2 , the minimum along p_1 , and construct the new direction

$$p_2 = -g_2 + \beta_{21} p_1 + \beta_{2o} p_o , \qquad (5.35)$$

such that p_2 satisfies the conditions

$$p_2^T (g_2 - g_1) = o , \quad p_2^T (g_1 - g_o) = o .$$

This guarantees that p_2 is conjugate to the two previous search directions p_1, p_o . In order to determine the constants β_{2o} and β_{21} ,we make use of some orthogonality relations. The perfect line search implies that $p_1^T g_2 = o$, but we have also $p_o^T g_2 = o$:

With $x_2 = x_1 + \alpha_1 p_1$ and (5.33a) we have

$$g_2 - g_1 = \alpha_1 G p_1 ,$$

and therefore

$$p_o^T g_2 = p_o^T g_1 + \alpha_1 p_o^T G p_1 = o ,$$

or $g_o^T g_2 = o$. We have further $g_o^T g_1 = o$, and from $p_1^T g_2 = (-g_1 + \beta_{1o} p_o)^T g_2 = o$ in addition $g_1^T g_2 = o$. The constants β_{21} and β_{2o} in (5.35) can easily be determined now and one finds

$$\beta_{21} = \frac{g_2^T g_2}{g_1^T g_1} , \quad \beta_{2o} = o .$$

This process can be continued leading to the general expression

$$p_k = -g_k + \sum_{i=o}^{k-1} \beta_{ki}\, p_i \ . \qquad (5.36)$$

Before we calculate the β_{ki} explicitly we need an orthogonality relation which will also be used in later chapters.

5.2.2. An important orthogonality relation

Let p_i , $i = o(1)k$, form a set of conjugate directions and

$$x_{i+1} = x_i + \alpha_i\, p_i \ , \quad i = o(1)k \ ,$$

where the α_i are determined by perfect line searches. Then:

$$p_i^T\, g_{k+1} = o \ , \qquad i = o(1)k \ , \ k = o(1)n-1 \qquad (5.37)$$

For $i=k$, (5.37) holds because of the perfect line search along p_k . For $i<k$:

$$x_{k+1} = x_{i+1} + \sum_{j=i+1}^{k} \alpha_j\, p_j \ , \quad i = o(1)k-1 \ , \quad k \geq 1 \ .$$

$$g_{k+1} = g_{i+1} + \sum_{j=i+1}^{k} \alpha_j\, G\, p_j \ .$$

Therefore we have

$$p_i^T\, g_{k+1} = p_i^T\, g_{i+1} + \sum_{j=i+1}^{k} \alpha_j\, p_i^T\, G\, p_j = o \ ,$$

where $p_i^T\, g_{i+1} = o$ because of the line search property, and $p_i^T\, G\, p_j = o$ because of the conjugacy property. This completes the proof of (5.37).

5.2.3. Determination of the coefficients β_{ki}

For the method of conjugate gradients one can show in particular that

$$g_i^T g_k = o \ , \quad i = o(1)k-1 \ , \quad k = 1(1)n-1 \ . \tag{5.38}$$

In section 5.2.1. this was found already for $k=1$ and $k=2$. From (5.36) we obtain

$$g_i = -p_i + \sum_{j=o}^{i-1} \beta_{ij} p_j \ , \qquad i = 1(1)k \ .$$

The directions $p_o,\ldots,p_k$ are conjugate by construction and $x_1,\ldots,x_k$ determined by perfect line searches. We can therefore apply (5.37):

$$g_i^T g_k = -p_i^T g_k + \sum_{j=o}^{i-1} \beta_{ij} p_j^T g_k = o \ , \quad i = 1(1)k-1 \ .$$

It remains to show $g_o^T g_k = o$, $k > o$. But $g_o^T g_k = -p_o^T g_k = o$, which follows also from (5.37).

Now we can readily calculate the coefficients β_{ki} appearing in (5.36). In order to ensure the conjugacy of the direction p_k we must satisfy the *construction equations*

$$p_k^T y_i = o \ , \quad i = o(1)k-1 \ .$$

With (5.36) this becomes

$$-g_k^T y_i + \sum_{j=o}^{k-1} \beta_{kj} p_j^T y_i = o \ , \quad i = o(1)k-1 \ . \tag{5.39a}$$

This set of equations for the unknowns β_{kj} can be simplified if we apply (5.38):

$$g_k^T y_i = o \ , \quad i = o(1)k-2 \ , \quad k \geq 2 \ ,$$

and (5.37):

$$p_j^T y_i = o \ , \quad j = o(1)i-1 \ , \quad i = 1(1)k-1 \ .$$

Moreover we can add the whole set of previous construction equations

$$p_j^T y_i = o \ , \quad i = o(1)j-1 \ , \quad j = 1(1)k-1 \ .$$

Hereafter (5.39a) reduces to

$$-g_k^T y_{k-1} \delta_{i,k-1} + \beta_{ki} p_i^T y_i = o \ , \quad i = o(1)k-1 \ , \qquad (5.39b)$$

and the final solution is (we set $\beta_k := \beta_{ki}$ for $i = k-1$):

$$\beta_{ki} = o \ , \quad i = o(1)k-2 \ ,$$

$$\beta_k = \frac{g_k^T y_{k-1}}{p_{k-1}^T y_{k-1}} \ . \qquad (5.4oa)$$

Therefore the method of conjugate gradients assumes the simple form

$$p_k = -g_k + \beta_k p_{k-1} \ , \quad k > o \ . \qquad (5.41)$$

Different representations of β_k are possible which are equivalent for quadratic functions and perfect line searches (see e.g. Dixon |16|). Thus (5.4oa) is known as the *Hestenes-Stiefel formula* |23|.

Another expression for β_k is due to Polak and Ribière |38|, who simplified the denominator of (5.4oa):

$$p_{k-1}^T y_{k-1} \equiv p_{k-1}^T (g_k - g_{k-1}) = -p_{k-1}^T g_{k-1}$$

$$= -(-g_{k-1} + \beta_{k-1} p_{k-2})^T g_{k-1} = g_{k-1}^T g_{k-1} \ ,$$

where the line search property along p_{k-1} and p_{k-2} has been used.
This gives the *Polak-Ribière formula*

$$\beta_k = \frac{g_k^T y_{k-1}}{g_{k-1}^T g_{k-1}} \ . \qquad (5.4ob)$$

Application of (5.38) to (5.4ob) finally yields the *Fletcher-Reeves formula* |19|, which already appeared in section 5.2.1. for k=1 and k=2 :

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} . \qquad (5.4oc)$$

Recent numerical results seem to indicate that (5.4ob) should be preferred (see the discussion at the end of this paragraph).

A fourth formula is due to *Daniel* |13|:

$$\beta_k = \frac{g_k^T A_k p_{k-1}}{p_{k-1}^T A_k p_{k-1}} , \qquad (5.4od)$$

where A_k is the Hessian matrix of second derivatives of f at x_k. For quadratic functions we have $A_k = G$, and with $s_{k-1} = \alpha_{k-1} p_{k-1}$ and $Gs_{k-1} = y_{k-1}$, (5.4od) becomes the Hestenes-Stiefel formula (5.4oa). The appearance of the matrix of second derivatives makes a practical application of (5.4od) less attractive.

5.2.4. Other approaches to the method of conjugate gradients

The derivation of the fundamental relation (5.41) presented in section 5.2.3. was based on the construction principle for conjugate directions from the knowledge of gradients only.

A completely different approach was suggested by Stoer and Bulirsch (|1o|, pp.263). They derive (5.41) from the minimization of the quadratic function

$$f(x_k + \mu_o g_o + \mu_1 g_1 + \ldots + \mu_k g_k)$$

with respect to $\mu_o, \ldots, \mu_k$, yielding $f(\bar{x}_{k+1})$. We want to show that $\bar{x}_{k+1} \equiv x_{k+1}$, where x_{k+1} is the point found by the conjugate gradient method.

From property vi) of paragraph 4.1. we know that x_{k+1} is the minimum in the subspace, which contains the point x_o and which is spanned by the conjugate directions $p_o,\ldots,p_k$. From (5.41) it follows that the p_k are linear combinations of $g_o,\ldots,g_k$, therefore the sets $\{p_i\}$ and $\{g_i\}$ span the same subspace. If we start at the common point x_o , the conjugate gradient method generates points which are identical to the points obtained if each iteration finds explicitly the minimum in the subspace which is spanned by all g_i available so far and which contains the minimum of the previous iteration.

We want to add a further derivation of (5.41) based on a Gram-Schmidt orthogonalization type procedure (see Beckmann | 4 |, and Beale | 3 |, pp. 39). From the general expression (5.36):

$$p_k = -g_k + \sum_{j=o}^{k-1} \beta_{kj}\, p_j \ ,$$

the conjugacy requirement of the new direction p_k to all previous (conjugate) directions leads to

$$p_k^T\, G\, p_i = o \ , \quad i = o(1)k-1 \ ,$$

or

$$-g_k^T\, G\, p_i + \beta_{ki}\, p_i^T\, G\, p_i = o \ , \quad i = o(1)k-1 \ . \qquad (5.42)$$

From $x_{i+1} = x_i + \alpha_i\, p_i$:

$$G\, p_i = \frac{1}{\alpha_i}\,(g_{i+1} - g_i) \ .$$

With (5.38) we obtain

$$g_k^T\, G\, p_i = o \ , \quad i = o(1)k-2 \ ,$$

and therefore from (5.42): $\beta_{ki} = o$, $i = o(1)k-2$, and finally for $i = k-1$:

$$\beta_k = \frac{g_k^T\,(g_k - g_{k-1})}{p_{k-1}^T\,(g_k - g_{k-1})} \ ,$$

which is again (5.4oa).

Instead of using (5.41) Rutishauser |45| suggested an alternative expression which we shall need in the following paragraph. The equivalence of both formulae is valid again only for quadratic functions and perfect line searches. Inserting (5.41) into $x_{k+1} = x_k + \alpha_k p_k$ yields

$$x_{k+1} = x_k + \alpha_k(-g_k + \beta_k p_{k-1})$$
$$= x_k + \alpha_k(-g_k + \beta_k(x_k - x_{k-1})/\alpha_{k-1}) \quad . \qquad (5.43)$$

With $h(\lambda) = f(x_k + \lambda p_k)$ we obtain $\lambda = \alpha_k$ from the line search condition $h'(\lambda) = o$, or

$$\alpha_k = - \frac{p_k^T G (x_k - x^*)}{p_k^T G p_k} \quad .$$

With (5.41) this becomes

$$\alpha_k = - \frac{(-g_k + \beta_k p_{k-1})^T g_k}{p_k^T G (-g_k + \beta_k p_{k-1})} = \frac{-g_k^T g_k}{p_k^T G g_k}$$

$$= \frac{-g_k^T g_k}{(-g_k + \beta_k p_{k-1})^T G g_k} \quad .$$

Es ist

$$g_k^T G p_{k-1} = g_k^T (g_k - g_{k-1})/\alpha_{k-1} = g_k^T g_k/\alpha_{k-1} \quad ,$$

and we obtain the recurrence relation for $1/\alpha_k$:

$$\frac{1}{\alpha_k} = \frac{g_k^T G g_k}{g_k^T g_k} - \frac{\beta_k}{\alpha_{k-1}} \quad , \quad k > o \quad , \qquad (5.44)$$

with the initial value

$$\frac{1}{\alpha_o} = \frac{g_o^T G\, g_o}{g_o^T g_o} .$$

This relation together with (5.43) is the Rutishauser result.

5.2.5. The reduction of $\| x_k - x^* \|$

Any minimization algorithm should necessarily reduce the function value at each iteration in order to be convergent. This, however, does not imply that x_k approaches the minimum x^* for each iteration. The method of conjugate gradients when applied to quadratic functions possesses the remarkable property that the euclidean norm $\| x_k - x^* \|$ decreases monotonously with k . Our derivation follows Beckmann |4|.

First we show that

$$p_i^T p_j > o . \qquad (5.45)$$

Successive application of (5.41) yields

$$p_i = - \{g_i + \beta_i\, g_{i-1} + \beta_i\, \beta_{i-1}\, g_{i-2} + \dots + \beta_i \dots \beta_1\, g_o\} ,$$

and with β_j from (5.4oc):

$$p_i = - \{g_i + \frac{g_i^T g_i}{g_{i-1}^T g_{i-1}}\, g_{i-1} + \frac{g_i^T g_i}{g_{i-2}^T g_{i-2}}\, g_{i-2} + \dots + \frac{g_i^T g_i}{g_o^T g_o}\, g_o\}$$

$$= -g_i^T g_i \sum_{\ell=o}^{i} \frac{1}{g_\ell^T g_\ell}\, g_\ell .$$

This together with the orthogonality relations (5.38) leads to

$$p_i^T p_j = \| g_i \|^2 \| g_j \|^2 \sum_{\ell=o}^{i} \frac{1}{\| g_\ell \|^2} > o , \quad i \leq j .$$

(5.45) means that once $p_o := -g_o$ is chosen all subsequent directions will lie in a cone around p_o with an angle less than 9o degrees.

Consider

$$\| x_k - x^* \|^2 = \| x_{k+1} - \alpha_k p_k - x^* \|^2$$

$$= \| x_{k+1} - x^* \|^2 + \alpha_k^2 \| p_k \|^2 + 2\alpha_k p_k^T (x^* - x_{k+1}) .$$

With

$$x^* = x_{k+1} + \sum_{j=k+1}^{n-1} \alpha_j p_j$$

we get

$$p_k^T (x^* - x_{k+1}) = \sum_{j=k+1}^{n-1} \alpha_j p_k^T p_j > o ,$$

because $p_k^T p_j > o$ from (5.45) and $\alpha_j > o$ follows from the descent property of p_j at the point x_j :

$$h'(x_j + \lambda p_j)|_{\lambda=o} = p_j^T g_j = (-g_j + \beta_j p_{j-1})^T g_j = -g_j^T g_j < o .$$

Thus we obtain

$$\| x_k - x^* \|^2 > \| x_{k+1} - x^* \|^2 ,$$

which proves that the sequence of points x_k generated by the method of conjugate gradients if applied to quadratic functions with perfect line searches tends monotonously towards the minimum x^* .

5.2.6. The method of conjugate gradients as an optimal process

Occasionally we find in the literature the method of conjugate gradients interpreted as an optimal process (see e.g. Stiefel |51|, Daniel |13|, Luenberger |31|). From this interpretation certain bounds for the convergence rate can be derived. Our presentation will mainly follow the one given by Luenberger (|31|, pp. 176).

We shall need a relation between the vectors p_k and g_o. For this purpose we want to derive a recurrence relation for the vector $(g_k^T, p_k^T)^T$. With

$$g_{k+1} = G(x_{k+1} - x^*) = G(x_k - x^* + \alpha_k p_k) = g_k + \alpha_k G\, p_k$$

we obtain from (5.41):

$$p_{k+1} = -g_{k+1} + \beta_{k+1} p_k = -g_k + \beta_{k+1} p_k - \alpha_k G\, p_k$$

and therefore

$$\begin{pmatrix} g_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} I & \alpha_k G \\ -I & \beta_{k+1} I - \alpha_k G \end{pmatrix} \begin{pmatrix} g_k \\ p_k \end{pmatrix} .$$

For $p_o = -g_o$, this recurrence relation leads to expressions of the form

$$g_k = \mathcal{L}(g_o, Gg_o, \ldots, G^k g_o) ,$$
$$p_k = \mathcal{L}(g_o, Gg_o, \ldots, G^k g_o) ,$$

where $\mathcal{L}(\cdot)$ denotes a linear combination of the arguments.

Insertion of the p_i into $x_k = x_o + \alpha_o p_o + \ldots + \alpha_{k-1} p_{k-1}$ yields an expression

$$x_k = x_o + P_{k-1}(G)\, g_o , \qquad (5.46)$$

where $P_k(\lambda)$ is a polynomial of degree k :

$$P_k(\lambda) = \sum_{i=o}^{k} \gamma_{k,i}\, \lambda^i .$$

As a measure for the deviation from the minimum we consider

$$E_k := f(x_k) - f(x^*) \ .$$

With (5.46) we obtain

$$E_k = \frac{1}{2}(x_o - x^* + P_{k-1}(G)g_o)^T G(x_o - x^* + P_{k-1}(G)g_o) \ . \quad (5.47)$$

As in (5.7a), we use the eigenvector decomposition

$$x_o - x^* = \sum_{i=1}^{n} c_{o,i} \, v_i \ .$$

Then (5.47) becomes

$$E_k = \frac{1}{2} \sum_{i=1}^{n} \{1 + \lambda_i P_{k-1}(\lambda_i)\}^2 \lambda_i \, c_{o,i}^2 \ . \quad (5.47')$$

Now consider the case where $P_k(\lambda)$ is replaced by an arbitrary polynomial

$$\bar{P}_k(\lambda) = \sum_{i=o}^{k} \bar{\gamma}_i \, \lambda^i \ .$$

Instead of (5.46) we obtain

$$\bar{x}_k = x_o + \bar{P}_{k-1}(G)g_o \ ,$$

where $\bar{x}_k$ is, like x_k, a point in the subspace which contains x_o and which is spanned by $g_o, Gg_o, \ldots, G^{k-1}g_o$ or likewise by $p_o, \ldots, p_{k-1}$. However, whereas x_k is the minimum of this subspace, this is no longer true for $\bar{x}_k$. Therefore we conclude that the method of conjugate gradients selects among all polynomials $\bar{P}_{k-1}(\lambda)$ the one which minimizes

$$\bar{E}_k(\bar{\gamma}_o, \ldots, \bar{\gamma}_{k-1}) = \frac{1}{2} \sum_{i=1}^{n} \{1 + \lambda_i \bar{P}_{k-1}(\lambda_i)\}^2 \lambda_i \, c_{o,i}^2 \quad (5.48)$$

with respect to $\bar{\gamma}_o, \ldots, \bar{\gamma}_{k-1}$. Because of this property, the method of conjugate gradients has been interpreted as an *optimal process*.

Inserting different polynomials $\bar{P}_{k-1}(\lambda)$ into (5.48) yields different upper bounds for E_k . A possible choice is given by

$$1 + \lambda \bar{P}_{k-1}(\lambda) = c_k \, T_k(\rho(\lambda)) \ , \tag{5.49}$$

where

$$\rho(\lambda) := \frac{\lambda_n + \lambda_1 - 2\lambda}{\lambda_n - \lambda_1} \ ,$$

and $T_k(\rho)$ is the Chebyshev polynomial of order k :

$$T_k(\rho) := \begin{cases} \cos\,(k \arccos \rho) \ , & |\rho| \le 1 \ , \\ \frac{1}{2}\{(\rho + \sqrt{\rho^2-1})^k + (\rho - \sqrt{\rho^2-1})^k\} \ , & |\rho| > 1 \ . \end{cases}$$

The constant c_k in (5.49) follows from $\lambda=o$:

$$\frac{1}{c_k} = T_k(\rho(o)) \ .$$

With $\rho(\lambda_1) = 1$, $\rho(\lambda_n) = -1$ we have

$$|T_k(\rho(\lambda_i))| \le 1 \ , \quad i = 1,\ldots,n \ .$$

With $\rho(o) > o$ and $\eta := \lambda_1/\lambda_n$ we find

$$T_k(\rho(o)) = \frac{1}{2(1-\eta)^k}\{(1+\sqrt{\eta})^{2k} + (1-\sqrt{\eta})^{2k}\} \ .$$

Inserting these results into (5.49) and using

$$E_o = \frac{1}{2} \sum_{i=1}^{n} \lambda_i \, c_{o,i}^2 \ ,$$

we obtain an upper bound for $\bar{E}_k$:

$$\bar{E}_k \le 4 \, \frac{(1-\eta)^{2k}}{\{(1+\sqrt{\eta})^{2k} + (1-\sqrt{\eta})^{2k}\}^2} \, E_o \ , \quad \eta := \frac{\lambda_1}{\lambda_n} \ ,$$

or finally

$$E_k \leq \bar{E}_k < 4 \frac{(\sqrt{\lambda_n} - \sqrt{\lambda_1})^{2k}}{(\sqrt{\lambda_n} + \sqrt{\lambda_1})^{2k}} E_o .$$

This result compares favourably with the corresponding result (5.17) for the optimum gradient method. It should be noted, however, that it still can be far too pessimistic, in particular we have always $E_n = o$.

5.2.7. Quadratic termination without perfect line searches

For quadratic functions perfect line searches do not require essentially more computational effort than imperfect line searches, we only have to set $x_{k+1} = x_k + \alpha_k p_k$ with

$$\alpha_k = - \frac{p_k^T g_k}{p_k^T G p_k} . \qquad (5.5o)$$

For non-quadratic functions the perfect line search is nearly impossible due to computational reasons. Still one would like to have an algorithm, which possesses the property of quadratic termination, because then the algorithm is expected to behave satisfactorily near the minimum, where the objective function can be approximated by a quadratic function. One essential goal of modern algorithms is to achieve quadratic termination without perfect line searches.

For the method of conjugate gradients Dixon |16| has shown how to obtain the minimum of a quadratic function in at most n iterations without explicit knowledge of the matrix G , which would be required if (5.5o) is to be applied. Here we want to present a slightly simplified version of this algorithm.

Let us assume that Dixon's "imperfect" algorithm has generated a sequence of directions $p_o,\ldots,p_{k-1}$ and points $x_o,\ldots,x_k$, which coincide with the corresponding quantities of the standard conjugate gradient method. Then the new search direction

$$p_k = -g_k + \beta_k p_{k-1} ,$$

where β_k must be chosen according to one of the formulae (5.4o), is again identical for both algorithms.

The *arbitrary step* $\bar{\alpha}_k$ along p_k yields $\bar{x}_{k+1} = x_k + \bar{\alpha}_k p_k$. Now the difference $\bar{s}_k := \bar{x}_{k+1} - x_k$ must be proportional to $s_k := x_{k+1} - x_k$:

$$\bar{s}_k = \Theta_k s_k , \tag{5.51a}$$

and with $Gs = y$ also

$$\bar{y}_k = \Theta_k y_k . \tag{5.51b}$$

The parameter Θ_k follows if we apply the perfect line search condition to (5.51b):

$$p_k^T \bar{y}_k = \Theta_k p_k^T (g_{k+1} - g_k) = -\Theta_k p_k^T g_k ,$$

or

$$\Theta_k = - \frac{p_k^T \bar{y}_k}{p_k^T g_k} .$$

The new point x_{k+1} and the gradient g_{k+1} follow from (5.51):

$$x_{k+1} = x_k + \frac{1}{\Theta_k} (\bar{x}_{k+1} - x_k) ,$$

$$g_{k+1} = g_k + \frac{1}{\Theta_k} (\bar{g}_{k+1} - g_k) .$$

We observe that this algorithm requires one more gradient evaluation $\bar{g}_n$ to determine Θ_{n-1}, and two more vectors $\bar{x}_{k+1}$, $\bar{g}_{k+1}$ to be stored.

Perry [37] suggested a conjugate gradient method which satisfies the perfect line search condition even for non-quadratic functions with inaccurate line searches. Let $x_{k+1} = x_k + \bar{\alpha}_k p_k$ be an arbitrary point along the direction p_k, and let g_{k+1} and s_k be defined as usual. We look for a point $\bar{x}_k$ such that a perfect line search along $\bar{s}_k = x_{k+1} - \bar{x}_k$ would have led to x_{k+1}.

With

$$\bar{s}_k = s_k - \frac{g_{k+1}^T s_k}{g_{k+1}^T g_{k+1}} g_{k+1}$$

we have indeed $g_{k+1}^T \bar{s}_k = o$, and from $\bar{x}_k = x_{k+1} - \bar{s}_k$ we can determine $\bar{g}_k$ and $\bar{y}_k = g_{k+1} - \bar{g}_k$. The new search direction is defined by

$$p_{k+1} = -g_{k+1} - \frac{g_{k+1}^T \bar{y}_k}{\bar{g}_k^T \bar{s}_k} \bar{s}_k ,$$

which is the Hestenes-Stiefel formula (5.4oa) with the old direction given by $\bar{s}_k$, and where $g_{k+1}^T \bar{s}_k = o$ has been used.

Unfortunately, Perry's *self correcting conjugate gradient* (SCCG) method does not possess the property of quadratic termination. Still his numerical results are most promising and indicate that a further investigation into this algorithm and the line search strategy used to determine $\bar{\alpha}_k$ seems to be worthwhile.

5.2.8. The method of Shah, Buehler and Kempthorne

The basic form of the algorithm of Shah et al |47| works for quadratic functions as follows. Generate a sequence of points $x_o, x_2, x_3, x_4, \ldots, x_{2n}$ such that

1) the vectors $x_{2k+1} - x_{2k}$, $k = 1(1)n-1$, are parallel to the planes π_{2j} , $j = o(1)k-1$, where π_ℓ is the hyperplane tangent to $f(x)$ at $x = x_\ell$,

2) the points x_{2k+2} , $k = 1(1)n-1$, lie on the line joining x_{2k-2} and x_{2k+1} . The point x_2 is arbitrary but should satisfy $f(x_2) < f(x_o)$.

This method is known as *Partan*, derived from "parallel tangents" which appear in the first condition.

The directions, which according to the second condition lead to the even-numbered points, are uniquely defined, whereas the directions of the first condition must be specified. One possibility is to choose the steepest descent directions $-g_{2k}$ at the even-numbered points. This variant is known as *Steepest Descent Partan*.

We note that according to the construction principle as given above, gradients are not necessary in the first instance, therefore Partan can be classified as a *method of conjugate directions* and as such should have been presented in chapter 4. But because only steepest descent Partan, which of course uses gradients, is of major interest this method belongs to the class of *gradient methods*.

Following Fletcher (in |2o|, pp. 83), we show that steepest descent Partan and the method of conjugate gradients behave identically when applied to quadratic functions with perfect line searches. Since Fletcher's proof is rather rudimentary ("and (5.4.5) can be obtained in a similar way by using the condition $g_{i+1}^T(z_i - x_{i-1}) = o$ ", p. 84), we want to derive the equivalence by induction.

In Fletcher's notation steepest descent Partan can be formulated as

$$\left.\begin{aligned} x_1 &= x_o - \mu_o\, g_o\ , \\ z_k &= x_k - \mu_k\, g_k \\ x_{k+1} &= z_k + \lambda_k (z_k - x_{k-1}) \end{aligned}\right\} \quad k = 1(1)n-1\ . \tag{5.52}$$

The μ_k, λ_k are obtained from perfect line searches. In order to show that the x_k of (5.52) coincide with the points generated by the method of conjugate gradients we observe first that $p_o = -g_o$ and therefore x_1 is the same for both methods.

Next we assume that x_k from (5.52) and all previous points are identical to the corresponding points of the method of conjugate gradients. Therefore x_k is related to x_{k-1} by

$$\begin{aligned} x_k &= x_{k-1} + \alpha_{k-1}\, p_{k-1}\ , \\ p_{k-1} &= -g_{k-1} + \beta_{k-1}\, p_{k-2}\ . \end{aligned}$$

From (5.52) we have

$$x_{k+1} = x_k - \mu_k\, (1 + \lambda_k)\, g_k + \lambda_k\, (x_k - x_{k-1})\ , \tag{5.53}$$

and μ_k follows from the perfect line search along $-g_k$ (see (5.5o)):

$$\mu_k = \frac{g_k^T\, g_k}{g_k^T\, G\, g_k}\ . \tag{5.54a}$$

In the same way, the line search along

$$z_k - x_{k-1} = x_k - x_{k-1} - \mu_k\, g_k = \alpha_{k-1}\, p_{k-1} - \mu_k\, g_k$$

yields

$$\lambda_k = \frac{-(\alpha_{k-1} p_{k-1} - \mu_k g_k)^T\, g_k}{(\alpha_{k-1} p_{k-1} - \mu_k g_k)^T\, (g_k - g_{k-1} - \mu_k\, G\, g_k)}\ .$$

The induction hypothesis allows to make use of

$$1)\quad p_{k-1}^T\, g_k = o\ ,$$

$$2)\quad g_{k-1}^T\, g_k = o\ ,$$

$$3)\quad p_{k-1}^T\, g_{k-1} = (-g_{k-1} + \beta_{k-1}\, p_{k-2})^T\, g_{k-1} = -g_{k-1}^T\, g_{k-1}\ .$$

Then λ_k becomes

$$\lambda_k = \frac{\mu_k\, g_k^T\, g_k}{(\mu_k^2\, g_k^T\, G\, g_k + \alpha_{k-1}\, g_{k-1}^T\, g_{k-1} - 2\mu_k\, g_k^T\, g_k)}\ ,$$

or with (5.54a):

$$\lambda_k = \frac{(g_k^T\, g_k)^2}{\alpha_{k-1}\, g_{k-1}^T\, g_{k-1} \cdot g_k^T\, G\, g_k - (g_k^T\, g_k)^2}\ .$$

Using the Fletcher-Reeves formula (5.4oc) for β_k, this can also be written as

$$\lambda_k = \frac{\beta_k\, g_k^T\, g_k}{(\alpha_{k-1}\, g_k^T\, G\, g_k - \beta_k\, g_k^T\, g_k)}\ . \qquad (5.54b)$$

Insertion of (5.54) into (5.53) yields

$$x_{k+1} - x_k = -\mu_k(1 + \lambda_k)\, g_k + \lambda_k\, \alpha_{k-1}\, p_{k-1}$$

$$= \frac{\alpha_{k-1}\, g_k^T\, g_k}{(\alpha_{k-1}\, g_k^T\, G\, g_k - \beta_k\, g_k^T\, g_k)}\, (-g_k + \beta_k\, p_{k-1})\ .$$

Application of the Rutishauser relation (5.44) finally leads to

$$x_{k+1} = x_k + \alpha_k\, p_k\ ,$$

which completes the proof by induction.

A disadvantage of the Partan method certainly is the fact that each iteration requires two line searches for the generation of one conjugate direction. On the other hand, each iteration provides one steepest descent step, which means that Partan is at least as good as the optimum gradient method.

5.2.9. Some numerical results

Table 5.2 illustrates the quadratic termination property of the method of conjugate gradients when applied to the Hilbert function introduced in section 5.1.5.

it.	n=2	n=3	n=4	n=5
1	o.5oQ-o2	o.2oQ-o1	o.44Q-o1	o.74Q-o1
2	o.34Q-63 (-33)	o.13Q-o4	o.95Q-o4	o.3oQ-o3
3	o.92Q-67 (-33)	o.43Q-58 (-24)	o.24Q-o7	o.26Q-o6
4		o.23Q-63 (-32)	o.11Q-5o (-17)	o.36Q-1o
5		o.59Q-68 (-33)	o.86Q-58 (-25)	o.44Q-43 (-1o)
6			o.27Q-62 (-29)	o.54Q-48 (-14)
7			o.14Q-67 (-34)	o.27Q-56 (-22)
8			o.1oQ-73 (-34)	o.57Q-63 (-29)
9				o.13Q-67 (-29)

Table 5.2 : Results of method of conjugate gradients applied to Hilbert function.

In all cases the quadratic termination can easily be observed when fourfold precision is used. In double precision (the numbers in brackets denote the exponents obtained in double precision) this property is evident only up to $n=4$. However, if the iteration numbers of Table 5.2 are compared with those of Table 5.1 , these rounding effects are completely negligible.

For non-quadratic functions and imperfect line searches the use of different expressions for β_k according to (5.4o) will in general lead to different results. For example, if two subsequent gradients, which in the quadratic case would always be orthogonal, are nearly equal $(g_k \approx g_{k-1})$, we would obtain $\beta_k \approx 1$ from (5.4oc), and $\beta_k \approx o$ from (5.4ob) . On the other hand, $\beta_k = o$ means $p_k = -g_k$, therefore the method of Polak and Ribière would in such occasions behave like the method of steepest descent, which is at least acceptable.

Fletcher and Reeves |19| found that their algorithm worked much better if after every n or $(n+1)$ iterations a new start is made along the direction of steepest descent. In section 4.6.1. such algorithms were termed "to operate in the restart or reset mode". In order to save the property of quadratic termination this restart cannot take place before n iterations have been completed. Restart methods can be very important to ensure convergence but this aspect will not be discussed here.

Easily available codes for the method of conjugate gradients are:

1) VA08AD : The Fletcher-Reeves method from Harwell Subroutine Library (HSL), implemented in 1964 and revised in 1972 (Fletcher |17|).

2) VA14AD : The Polak-Ribière method from HSL, with an improved line search strategy as against the one used in VA08AD, implemented in 1975 (Powell |42|).

3) DFMCG : The Fletcher-Reeves method from IBM Scientific Subroutine Package (SSP).

VA08AD uses the line search strategy as described in section 2.3.3. VA14AD was written by Powell |43| after having encountered great difficulties with VA08AD, when a very large problem (n=165) had to be solved. It should be noted at this point that the powerful variable metric methods discussed in the following chapters need to store a matrix of size $\frac{1}{2}$ n(n+1) , whereas the method of conjugate gradients requires only the storage of three arrays of size n each.

In Table 5.3 we list the results obtained with the three conjugate gradient codes VA08AD, VA14AD and DFMCG, when applied to the problems described in paragraph 4.7. We also refer to this paragraph for all other details, except that EFE stands for "equivalent function evaluations", meaning that every gradient call is treated as n function calls.

VA14AD is in *all* cases superior to the Fletcher-Reeves codes, in some cases even by an order of magnitude; DFMCG seems to be slightly inferior to VA08AD.

Comparing Table 5.3 and Table 4.1 , some interesting properties become clear. Let us first consider criterion I. In almost all cases the final function values of Table 4.1 are lower than those of Table 5.3 . However, the CPU time, which is nearly constant for the conjugate gradient methods, increases linearly with n for the conjugate direction methods (see also Fig. 5.1o of | 2 |) and is considerably higher for the high dimensional problems than in Table 5.3 . Now we know (see e.g. Hillstrom |24|) that the concept of equivalent function evaluations is too pessimistic, in general much less than the equivalent of n function evaluations is necessary to compute the gradient, whenever the gradient is known analytically. For this reason it is not possible to draw any significant conclusions for the two groups of algorithms from the information of criterion I.

Table 5.3: Comparison of three conjugate gradient codes.

Probl. ($f(x^*)$)	Code	EFE	f_{final} (2oo EFE)	T	EFE	T ($1o^{-2}$)	EFE	T ($1o^{-6}$)	EFE	T ($1o^{-1o}$)
3.1	VAO8AD	2oo	49.•••	o.57	72	o.1o	112	o.o7	14o	o.13
(49.61...)	VA14AD	152	49.•••	o.42	52	o.24	92	o.39	136	o.29
n=3	DFMCG	2oo	49.•••	o.53	96	o.27	144	o.29	2o8	o.57
3.2	VAO8AD	2o3	98.46	o.55	476	1.21	1869	4.83	2338	5.69
(54.39...)	VA14AD	2o3	54.81	o.51	2o3	o.53	56o	1.21	693	1.97
n=6	DFMCG	2o3	1o4.75	o.54	357	o.9o	1862	4.85	3248	8.o2
3.3	VAO8AD	2oo	89.23	o.47	356o	8.66	5o.o5o5		-	
(5o.o5...)	VA14AD	2oo	86.59	o.52	264o	6.49	485o	11.78	682o	16.49
n=9	DFMCG	2oo	1o9.oo	o.49	569o	13.9o	5o.o592		-	
3.4	VAO8AD	2o8	84.67	o.51	2o41	4.89	8658	2o.25	215o2	5o.43
(47.92...)	VA14AD	221	81.31	o.52	1729	4.17	62o1	14.79	15834	37.61
n=12	DFMCG	2o8	222.o9	o.5o	2oo2	4.76	8268	19.32	1599o	37.52
3.5	VAO8AD	2o8	236.6o	o.44	23168	54.44	42.631		-	
(42.4o...)	VA14AD	2o8	157.o5	o.55	672o	15.94	28976	66.48	42.4oo66	
n=15	DFMCG	2o8	28o.93	o.51	43.2o		-			
3.6	VAO8AD	2o9	2o8.74	o.5o	66.82		-		-	
(39.27...)	VA14AD	2o9	179.92	o.48	56.36		-		-	
n=18	DFMCG	2o9	275.69	o.48	63.o3		-		-	

(criterion I) (criterion II)

(Small differences in the results for VAO8AD as against those in | 2 | are presumably due to changes within the computer environment, e.g. a change of operating systems from MVT to MVS. This demonstrates the difficulty to reproduce numerical results, even for the same type of machine, over a longer period of time.)

The situation is quite different when we consider criterion II. Take for example the fourth problem. VA04AD needs 6.6 , 9.7 and 1o.8 seconds to reach the three levels, the corresponding values for VA14AD are 4.2 , 14.8 and 37.6 seconds. Whereas the codes of Table 4.1 clearly exhibit superlinear convergence, the codes of Table 5.3 seem to converge in most cases only linearly or weakly superlinearly.

We anticipate that even better results can be obtained with variable metric methods, as will be shown in chapter 8. For comparison we quote the figures for VA13AD (with derivatives): 1.47 , 2.42 and 2.58 seconds, and for VA1oAD (using numerical derivatives): 2.19 , 3.86 and 4.14 seconds. These figures can even be reduced in most cases, if we choose a more sophisticated initial matrix H_o than it was done here with $H_o = I$. However, these figures do not say anything about reliability, robustness or stability, desirable properties which express a certain guarantee that a code should be insensitive against poor choices of initial points x_o or rounding errors, and should not terminate prematurely in unwanted local minima, as is often the case with variable metric methods.

References

|1| Akaike, H.: On a Successive Transformation of Probability Distribution and Its Application to the Analysis of the Optimum Gradient Method. Ann. Inst. Statist. Math., Tokyo, 11 (1959) 1-16.

|2| Amadori, R., Mika, K. and v. Studnitz, I.: A Comparison of Performance of Unconstrained Minimization Algorithms for the Solution of Special Nonlinear Least Squares Problems, Report Jül-1277, Kernforschungsanlage Jülich, März 1976.

|3| Beale, E.M.L.: A Derivation of Conjugate Gradients. In: Numerical Methods for Nonlinear Optimization. F.A. Lootsma (ed.), Academic Press, London and New York, 1972, 39-43.

|4| Beckmann, F.S.: The solution of linear equations by the conjugate gradient method. In: Mathematical Methods for Digital Computers I. A. Ralston and H.S. Wilf (eds.), John Wiley, New York, 1962, 62-72.

|5| Beltrami, E.J. and Indusi, J.P.: An Adaptive Random Search Algorithm for Constrained Minimization. IEEE Trans. on Comp., C-21 (1972) 1004-1008.

|6| Box, M.J., Davies, D. and Swann, W.H.: Nonlinear Optimization Techniques. ICI Monograph 5. Oliver and Boyd, Edinburgh, 1969.

|7| Box, M.J.: A new method of constrained optimization and a comparison with other methods. Comp. J., 8 (1965) 42-52.

|8| Brent, R.P.: Algorithms for Minimization without Derivatives. Prentice-Hall, Englewood Cliffs, N.J., 1973.

|9| Brodlie, K.W.: A New Direction Set Method for Unconstrained Minimization without Evaluating Derivatives. J. Inst. Math. Appl., 15 (1975) 385-396.

|1o| Bulirsch, R. and Stoer, J.: Einführung in die Numerische Mathematik II. Springer, Berlin-Heidelberg-New York, 1973.

|11| Burhardt, K.K.: An Adaptive Search Optimization Algorithm. IEEE Trans. on Comp., C-23 (1974) 89o-897.

|12| Cauchy, A.: Méthode générale pour la résolution des systèmes d'équations simultanées. Comptes Rendus 25 (1847) 536.

|13| Daniel, J.W.: The Conjugate Gradient Method for Linear and Nonlinear Operator Equations. SIAM J. Numer. Anal. 4 (1967) 1o-26.

|14| Davidon, W.C.: Variable Metric Method for Minimization. AEC Research and Development Report, Argonne National Laboratory, ANL-599o (rev.), 1959.

|15| Davies, D.: Some Practical Methods of Optimization. In: Integer and Nonlinear Programming, North-Holland, Amsterdam, 197o, 87-117.

|16| Dixon, L.C.W.: Conjugate Gradient Algorithms: Quadratic Termination without Linear Searches. J. Inst. Maths. Applics, 15 (1975) 9-18.

|17| Fletcher, R.: A FORTRAN Subroutine for Minimization by the Method of Conjugate Gradients. Report AERE-R7o73, Harwell, 1972.

|18| Fletcher, R.: FORTRAN Subroutines for Minimization by quasi-Newton Methods. Report AERE-R7125, Harwell, 1972.

|19| Fletcher, R. and Reeves, C.M.: Function minimization by conjugate gradients. Comp. J., 7 (1964) 149-154.

|2o| Fletcher, R.: Conjugate Direction Methods. In: Numerical Methods for Unconstrained Optimization. W. Murray (ed.), Academic Press, London and New York, 1972, 73-86.

|21| Fletcher, R.: Function minimization without evaluating derivatives - a review. Comp. J., 8 (1965) 33-41.

|22| Goldstein, A.A.: Cauchy's method of minimization. Num. Mathem., 4 (1962) 146-15o.

|23| Hestenes, M.R. and Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. N.B.S. 49 (1952) 4o9-436.

|24| Hillstrom, K.E.: A Simulation Test Approach to the Evaluation of Nonlinear Optimization Algorithms. ACM TOMS 3 (1977) 3o5-315.

|25| Himmelblau, D.M.: Applied Nonlinear Programming. McGraw-Hill Book Comp., N.Y., 1972.

|26| Hooke, R. and Jeeves, T.A.: "Direct Search" Solution of Numerical and Statistical Problems, J.A.C.M., 8 (1961) 212-229.

|27| Kiefer, J.: Sequential Minimax Search for a Maximum. Proc. Am. Math. Soc. 4 (1953) 5o2-5o6.

|28| Kowalik, J. and Osborne, M.R.: Methods for Unconstrained Optimization Problems. Am. Elsevier, New York, 1968.

|29| Kutzbach, K., Mika, K. and Wingerath, K.: A FORTRAN Version of Brent's Derivative free Algorithm. Kernforschungsanlage Jülich, 1978 (unpublished).

|3o| Lawrence, J.P. and Steiglitz, K.: Randomized Pattern Search. IEEE Trans. on Comp., C-21 (1972) 382-385.

|31| Luenberger, D.G.: Introduction to Linear and Nonlinear Programming. Addison-Wesley, Reading, Ma., 1973.

|32| Miranker, W.L. and Chazan, D.: A Nongradient and Parallel Algorithm for Unconstrained Minimization. SIAM J. Control, 8 (197o) 2o7-217.

|33| Murray, W.: Fundamentals. In: Numerical Methods for Unconstrained Optimization. W. Murray (ed.), Academic Press, London-N.Y., 1972, 1-12.

|34| Nelder, J.A. and Mead, R.: A simplex method for function minimization. Comp. J., 7 (1965) 3o8-313.

|35| Olsson, D.M. and Nelson, L.S.: The Nelder-Mead Simplex Procedure for Function Minimization. Technometrics, 17 (1975) 45-51.

|36| Oren, S.S. and Luenberger, D.G.: Self-Scaling Variable Metric (SSVM) Algorithms. Part I: Criteria and Sufficient Conditions for Scaling a Class of Algorithms. Managem. Science, 2o (1974) 845-862.

|37| Perry, A.: A Self Correcting Conjugate Gradient Algorithm, Int. J. Computer Math; Section B, 6 (1978) 327-333.

|38| Polak, E. and Ribière, G.: Note sur la Convergence de méthodes de directions conjugées, Revue Francaise d'Automatique, Informatique et Recherche Opérationelle, 3 Série R (1969), 35-43.

|39| Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comp. J., 7 (1964) 155-162.

|4o| Powell, M.J.D.: Some Global Convergence Properties of a Variable Metric Algorithm for Minimization without Exact Line Searches. Report CSS 15, AERE Harwell, April 1975.

|41| Powell, M.J.D.: A View of Unconstrained Minimization Algorithms That Do Not Require Derivatives. ACM Trans. on Mathem. Software, 1 (1975) 97-1o7.

|42| Powell, M.J.D.: Restart Procedures for the Conjugate Gradient Method. Report CSS 24, AERE Harwell, November 1975.

|43| Powell, M.J.D.: Restart Procedures for the Conjugate Gradient Method. Mathem. Progr., 12 (1977) 241-254.

|44| Rosenbrock, H.H.: An Automatic Method for finding the Greatest or Least Value of a Function. Comp. J., 3 (196o) 175-184.

|45| Rutishauser, H.: Theory of Gradient Methods. In: Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems. M. Engeli, Th. Ginsburg, H. Rutishauser and E. Stiefel (eds.), Birkhäuser, Basel, 1959, 24-49.

|46| Schrack, G. and Borowski, N.: An Experimental Comparison of Three Random Searches. In: Numerical Methods for Nonlinear Optimization. F.A. Lootsma (ed.), Academic Press, London and New York, 1972, 137-147.

|47| Shah, B.V., Buehler, R.J. and Kempthorne, O.: Some Algorithms for Minimizing a Function of Several Variables. SIAM J., 12 (1964) 74-92.

|48| Smith, C.S.: The automatic computation of maximum likelihood estimates. NCB Scient. Dept. Report SC 846 /MR/4o, 1962.

|49| Spendley, W.: Nonlinear Least Squares Fitting using a Modified Simplex Minimization Method. In: Optimization. R. Fletcher (ed.), Academic Press, London and New York, 1969, 259-27o.

|5o| Spendley, W., Hext, G.R. and Himsworth, F.R.: Sequential application of simplex designs in optimization and Evolutionary Operation, Technometrics, 4 (1962) 441-461.

|51| Stiefel, E.: Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme. Comment. Math. Helv. 29 (1955) 157-179.

|52| Sutti, C.: Remarks on Conjugate Direction Methods for Minimization without Derivatives. In: Towards global optimization. L.C.W. Dixon and G.P. Szegö (eds.), North-Holland, Amsterdam, 1975, 29o-3o4.

|53| Sutti, C.: A New Method for Unconstrained Minimization without Derivatives. In: Towards global optimisation. L.C.W. Dixon and G.P. Szegö (eds.), North-Holland, Amsterdam, 1975, 277- 289.

|54| Wilde, D.J.: Optimum Seeking Methods. Prentice-Hall, Englewood Cliffs, N.J., 1964.

|55| Zangwill, W.I.: Minimizing a function without calculating derivatives. Comp. J., 1o (1967) 293-296.

Acknowledgment

This work is based on a seminar which was given by the authors during the second term of 1976 at the Departamento de Informática of Pontifícia Universidade Católica (PUC), Rio de Janeiro. The authors wish to thank the director of the department, Prof. Carlos José Pereira de Lucena for his continuous support, and Prof. P. Albrecht for enabling this work and for his most valuable critical remarks and suggestions which greatly helped to improve this report. One of the authors (K.M.) wishes to thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Gesellschaft für Mathematik und Datenverarbeitung Bonn (GMD) and KFA Jülich for making this visit possible. Programming assistance by Frau Ursula Funk and Frau Karin Meuser is greatly acknowledged. The authors want to express their sincere gratitude to Frau Magda Kötter for her care and patience in typing this difficult matter.